



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

# **Análise Preditiva do Desempenho Acadêmico de Alunos de Graduação da UnB Utilizando Mineração de Dados**

Gabriel Ferreira Silva

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Ciência da Computação

Orientador  
Prof. Dr. Marcelo Ladeira

Brasília  
2017



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

# **Análise Preditiva do Desempenho Acadêmico de Alunos de Graduação da UnB Utilizando Mineração de Dados**

Gabriel Ferreira Silva

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Marcelo Ladeira (Orientador)  
CIC/UnB

Prof.a Dr.a Letícia Lopes Leite    Prof.a Dr.a Maria de Fátima Ramos Brandão  
CIC/Universidade de Brasília        CIC/Universidade de Brasília

Prof. Dr. Thiago de Paulo Faleiros  
CIC/Universidade de Brasília

Prof. Dr. Rodrigo Bonifácio  
Coordenador do Bacharelado em Ciência da Computação

Brasília, 14 de julho de 2017

# Dedicatória

Dedico este trabalho à minha família e aos meus amigos: vocês fazem a vida valer a pena!

# Agradecimentos

Agradeço a Deus, por ter me dado saúde, disposição, uma excelente família e ótimos amigos. Agradeço as pessoas que compõem o movimento Software Livre, que desenvolvem ótimos produtos e servem de inspiração para mim. Agradeço a UnB pela possibilidade de aprender com pessoas qualificadas e pelas oportunidades disponíveis. Agradeço ao pessoal do departamento, pelo clima de amizade que torna cursar ciência da computação uma atividade edificante e divertida ao mesmo tempo. Agradeço ao meu orientador, Ladeira, pela contínua disposição em me ajudar durante os dois anos que trabalhamos juntos. Agradeço a minha família e aos meus amigos por estarem ao meu lado, tanto nos momentos bons quanto nos momentos ruins: vocês fazem a vida valer a pena! Se alguém referido nesses agradecimentos tiver lido até aqui, basta me avisar e eu pago um brigadeiro como agradecimento por tudo!

# Resumo

A evasão de alunos de graduação na Universidade de Brasília (UnB) traz consequências acadêmicas, sociais e econômicas negativas. A UnB notifica aqueles alunos que estão em risco de serem desligados e que precisam cumprir condição para evitar o desligamento. Esses alunos em condição estabelecem um plano que devem cumprir, sob a orientação de um docente do curso do aluno. Pensando em melhorar tal abordagem, a pesquisa em questão objetiva a concepção de um sistema previsor capaz de indicar quais alunos estão com maior risco de não conseguirem formar. Desse modo, o sistema permitiria a UnB agir antes de um aluno entrar em condição e agir de acordo com o risco de evasão apresentado por cada aluno. Para o desenvolvimento do sistema previsor, dados descaracterizados de alunos de graduação de cursos da área de computação que ingressaram de 2000 até 2016 e já saíram da universidade foram utilizados. Algoritmos de aprendizagem de máquina foram usados (*Naive Bayes*, ANN, SVR, Regressor Linear e Random Forests) para induzir modelos que tiveram seu desempenho analisado. Os modelos obtidos testados tiveram, em geral, bom desempenho. O melhor desempenho foi para modelos induzidos com regressão linear. Os resultados obtidos apontam a viabilidade da utilização de mineração de dados para análise preditiva de alunos em risco de evasão na UnB nos cursos da área de computação. Como a metodologia utilizada não empregou nenhum conceito específico dessa área do conhecimento, pode-se usá-la para outros cursos de graduação da UnB.

**Palavras-chave:** análise preditiva, data mining, alunos em risco de evasão, UnB

# Abstract

The University of Brasília (UnB) suffers from a problem of student drop out, which has academic, economic and social negative consequences. UnB notifies its students that are in risk of dropping out and need to fulfill conditions to avoid dropping out. This students in conditions establish a plan they must accomplish, under the guidance of a professor of the same course of the student. The goal of the present research is the development of a predictive system capable of indicating the risk of a student dropping out. This way, UnB could act before it's became late and also act according to the risk presented by a student. For the development of the predictive system, data of undergraduate students from computer science related courses that entered and left UnB from 2000 to 2016 were used. The data do not contain student identification. Machine learning algorithms were used and their performance was evaluated. Algorithms included were Naive Bayes, ANN, SVR, Linear Regressor and Random Forests). Machine learning algorithms got, in general, good performance. The best performance came from the linear regressor. Results obtained indicate potential in using machine learning to predict the risk of students dropping out of university for the courses related to computer science. Because the methodology did not use any concept from this area of knowledge, this approach can be used for other courses.

**Keywords:** predictive analysis, machine learning, drop out, UnB

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Definição do Problema . . . . .	1
1.2	Proposta de Solução . . . . .	1
1.3	Objetivos . . . . .	2
1.4	Organização do Documento . . . . .	3
<b>2</b>	<b>Fundamentação Teórica</b>	<b>4</b>
2.1	Evasão nas Universidades e Especificidades da UnB . . . . .	4
2.1.1	Procedimentos da UnB Para Alunos com Risco de Evadir . . . . .	4
2.1.2	O Índice de Rendimento Acadêmico . . . . .	5
2.2	CRISP-DM . . . . .	6
2.2.1	Entendimento do Negócio . . . . .	7
2.2.2	Entendimento dos Dados . . . . .	7
2.2.3	Preparação dos Dados . . . . .	7
2.2.4	Modelagem . . . . .	8
2.2.5	Avaliação . . . . .	8
2.2.6	Implantação . . . . .	8
2.3	Algoritmos de Aprendizagem de Máquina . . . . .	8
2.3.1	Árvores de Decisão e Random Forest . . . . .	9
2.3.2	Regressão Linear . . . . .	9
2.3.3	SVR . . . . .	10
2.3.4	ANN . . . . .	11
2.3.5	Naive Bayes . . . . .	12
2.3.6	ZeroR . . . . .	12
2.4	Métricas Para Avaliação de Desempenho dos Modelos . . . . .	13
<b>3</b>	<b>Metodologia</b>	<b>14</b>
3.1	Levantamento do Estado da Arte . . . . .	14
3.2	Obtenção e Utilização dos Dados . . . . .	15

3.3	Seleção Preliminar de Atributos . . . . .	16
3.4	Eliminação de Atributos Devido a Missing Values . . . . .	17
3.5	Eliminação de Outliers . . . . .	17
3.6	Estatística Descritiva . . . . .	19
3.6.1	Mudança na Base de Dados . . . . .	19
3.6.2	Mudança nos Valores de Atributos . . . . .	20
3.6.3	Gráficos de Barra e Histogramas . . . . .	22
3.7	Eliminação de Atributos Relacionados ou Irrelevantes . . . . .	41
3.8	Divisão em Treino e Teste . . . . .	41
3.9	A Divisão em Semestres . . . . .	41
3.10	Algoritmos de Aprendizagem de Máquina Estudados e Retroalimentação . .	42
3.11	Ajuste de Parâmetros . . . . .	42
3.11.1	Ajuste de Parâmetros Para Redes Neurais . . . . .	43
3.11.2	Ajuste de Parâmetros Para SVR . . . . .	43
3.11.3	Estimativa de Parâmetros Para Naive Bayes . . . . .	43
3.12	Avaliação de Desempenho . . . . .	43
<b>4</b>	<b>Análise dos Resultados</b>	<b>45</b>
4.1	Configurações Obtidas Para Modelos de Aprendizagem de Máquina . . . . .	45
4.1.1	Configurações da ANN . . . . .	45
4.1.2	Configurações da SVR . . . . .	46
4.1.3	Configuração do Naive Bayes . . . . .	46
4.2	Desempenho dos Modelos de Aprendizagem de Máquina . . . . .	47
<b>5</b>	<b>Conclusão e Trabalhos Futuros</b>	<b>49</b>
5.1	Conclusão . . . . .	49
5.2	Trabalhos Futuros . . . . .	50
	<b>Referências</b>	<b>51</b>



# Lista de Figuras

2.1	Crítérios Para Desligamento na UnB . . . . .	5
2.2	Árvore de Decisão Para a Prática de Tênis . . . . .	9
2.3	Exemplo de SVM com Kernel Linear . . . . .	10
2.4	Exemplo de SVM com Kernel RBF . . . . .	10
2.5	Camadas de uma Rede Neural . . . . .	11
2.6	Diagrama do Classificador Probabilístico Naive Bayes . . . . .	12
3.1	Gráfico de Barra para Atributo Raça . . . . .	18
3.2	Gráfico de Barra para Atributo Tipo da Escola . . . . .	18
3.3	Gráfico de Barra de Todos os Alunos, para o Atributo Forma de Ingresso .	21
3.4	Gráfico de Barra para Atributo Forma de Saída . . . . .	22
3.5	Atributo Idade, Conforme os Diferentes Modelos . . . . .	23
3.6	Atributo Curso, Conforme os Diferentes Modelos . . . . .	25
3.7	Atributo Sexo, Conforme os Diferentes Modelos . . . . .	26
3.8	Atributo Cota, Conforme os Diferentes Modelos . . . . .	27
3.9	Atributo Tipo da Escola, Conforme os Diferentes Modelos . . . . .	28
3.10	Atributo Forma de Ingresso, Conforme os Diferentes Modelos . . . . .	30
3.11	Atributo Forma de Saída, Conforme os Diferentes Modelos . . . . .	31
3.12	Atributo Quantidade de Créditos, Conforme os Diferentes Modelos . . . . .	32
3.13	Atributo em condição, Conforme os Diferentes Modelos . . . . .	33
3.14	Atributo Taxa de Trancamento, Conforme os Diferentes Modelos . . . . .	34
3.15	Atributo Taxa de Falhas, Conforme os Diferentes Modelos . . . . .	35
3.16	Atributo Taxa de Aprovação em Matérias Difíceis, Conforme os Diferentes Modelos . . . . .	36
3.17	Atributo Taxa de Melhora, Conforme os Diferentes Modelos . . . . .	37
3.18	Atributo Média do Período, Conforme os Diferentes Modelos . . . . .	38
3.19	Atributo Taxa de Aprovação, Conforme os Diferentes Modelos . . . . .	39
3.20	Atributo Posição, Conforme os Diferentes Modelos . . . . .	40
3.21	Diagrama Explicativo Mostrando Processo de Avaliação do Desempenho . .	44

# Lista de Tabelas

2.1	Equivalência entre Menção Obtida por um Aluno e Nota Final na Disciplina	6
3.1	Percentagem de Alunos Capazes de Formar por Curso . . . . .	20
3.2	Percentagem de Alunos Capazes de Formar por Idade . . . . .	20
4.1	Melhor Escolha de Parâmetros Para ANN . . . . .	45
4.2	Melhor Escolha de Parâmetros Para SVR . . . . .	46
4.3	Melhor Escolha de Parâmetros Para Naive Bayes . . . . .	46
4.4	F-measure Média por Modelo . . . . .	47

# Lista de Abreviaturas e Siglas

**ANN** Artificial Neural Networks.

**CRISP-DM** Cross Industry Standard Process for Data Mining.

**FT** Faculdade de Tecnologia.

**IE** Instituto de Ciências Exatas.

**IRA** Índice de Rendimento Acadêmico.

**SIGRA** Sistema de Informação da Graduação.

**SVR** Support Vector Machine for Regression.

**UnB** Universidade de Brasília.

# Capítulo 1

## Introdução

Neste capítulo, faz-se uma introdução do trabalho. A definição do problema é feita e a proposta de solução é apresentada. Em seguida, descrevem-se os objetivos traçados e explica-se a estrutura geral do documento em questão.

### 1.1 Definição do Problema

A evasão de alunos de graduação nas universidades brasileiras já foi estudada por diversos autores [1] [2], tratando-se de um problema que traz desperdícios acadêmicos, sociais e econômicos. A Universidade de Brasília (UnB) não é exceção, sendo afetada significativamente pelo problema <sup>1</sup>.

A UnB notifica aqueles alunos que estão em risco de serem desligados e que precisam cumprir condição para evitar o desligamento. Esses alunos em condição estabelecem um plano que devem cumprir, sob a orientação de um docente do curso do aluno. Essa abordagem, entretanto, apresenta problemas:

- Os alunos (uma amostra muito diversificada) são classificados em apenas dois grupos.
- A UnB age apenas quando o aluno já se encontra em condição: pode já ser tarde demais.

### 1.2 Proposta de Solução

Essa pesquisa é acerca do desempenho acadêmico de alunos da UnB em sentido amplo. Segundo [4], o desempenho acadêmico definido em sentido amplo divide-se em três catego-

---

<sup>1</sup>A UnB teve um prejuízo com evasão estimado em 95,6 milhões, segundo o Correio Brasiliense na reportagem do dia 10/10/2015 “Evasões na Universidade de Brasília causam prejuízo de R\$ 95mi”[3]

rias: êxito, atraso e abandono. Nessa pesquisa, êxito, atraso e abandono são considerados como graduar, migrar e evadir respectivamente.

A pesquisa aqui descrita justifica-se como uma tentativa de melhorar a abordagem atual da UnB para o problema da evasão. Propõe-se utilizar dados passados para a criação de um sistema previsor que seja capaz de identificar alunos em risco de serem desligados. O sistema previsor fornece 3 valores positivos  $v_1$ ,  $v_2$ ,  $v_3$  que somam 1 e indicam respectivamente a chance do aluno se graduar, ser desligado (evadir) e migrar de curso.

Valores mais próximos de 0 indicam um baixo risco do evento (graduar, evadir ou migrar) em questão acontecer, enquanto valores mais próximos de 1 indicam um alto risco. Por exemplo, um valor de  $v_1$  próximo de 1 indica que o aluno tem grande chance de se graduar, ao passo que um valor de  $v_3$  próximo de 0 indica que o aluno tem risco baixo de migrar de curso.

Os dados utilizados são dados descaracterizados de alunos de graduação da UnB, contendo tanto informações de perfil quanto a forma de ingresso e as menções nas matérias da UnB.

Potencialmente, o sistema previsor permitirá a UnB agir antes de um aluno entrar em condição. Outra vantagem será a possibilidade de agir de acordo com o risco de evasão apresentado por cada aluno. Ou seja, a proposta de solução permitirá agir com antecedência e flexibilidade.

## 1.3 Objetivos

O objetivo da pesquisa é investigar a aplicação de técnicas de mineração de dados para o desenvolvimento de modelos de predição da conclusão do curso por alunos de graduação da UnB. Os objetivos específicos são:

- Avaliar a viabilidade dessa abordagem para uma área mais restrita do conhecimento para, em uma segunda fase, aplicar a metodologia desenvolvida para os demais cursos de graduação ofertados pela UnB.
- Induzir modelos que indiquem a possibilidade do aluno concluir, evadir ou migrar de curso.

Na presente etapa dessa pesquisa é investigada apenas a indução de modelos preditores para os cursos da área de computação ofertados pela UnB, ou seja: Ciência da Computação, Engenharia de Computação, Engenharia de Controle e Automação, Engenharia de Software, Engenharia de Redes e Licenciatura em Computação. O curso de Engenharia

de Controle e Automação também é chamado de Engenharia Mecatrônica e será assim denotado no restante desse documento.

Não é objetivo da pesquisa discutir as causas da evasão ou de migração de curso.

## **1.4 Organização do Documento**

Descreve-se a seguir a organização do documento. No Capítulo 2 explica-se a fundamentação teórica necessária para a compreensão da pesquisa em questão. Já no Capítulo 3, descreve-se a metodologia adotada na pesquisa. No Capítulo 4 apresentam-se e discutem-se os resultados obtidos. Finalmente, no Capítulo 5, apresenta-se a conclusão e apontam-se sugestões para trabalhos futuros. Apresenta-se a parte de estatística descritiva separadamente no apêndice A.

# Capítulo 2

## Fundamentação Teórica

Neste capítulo, descreve-se a fundamentação teórica necessária para compreender a pesquisa. Assim, nas seções seguintes explica-se a problemática da evasão nas universidades públicas e as especificidades da UnB, de modo a fornecer a base necessária para compreensão de alguns termos usados nessa pesquisa. Em seguida, comenta-se sobre o modelo de referência para mineração de dados CRISP-DM, que foi utilizado nessa pesquisa. Depois, os algoritmos de mineração de dados utilizados na pesquisa são explicados. Por fim, a *F-measure* é definida e a escolha dessa como métrica para avaliar os modelos induzidos é justificada.

### 2.1 Evasão nas Universidades e Especificidades da UnB

A evasão nas universidades nacionais é um problema que traz desperdícios acadêmicos, sociais e econômicos, tendo sido estudada por diversos autores [1] [5]. Apesar disso, o conceito de evasão pode variar de acordo com a pesquisa [5], já que pode-se considerar evasão do curso, evasão da instituição ou evasão do sistema educacional. Essa pesquisa foca na evasão de curso.

A UnB possui particularidades importantes tanto em relação às notas dos alunos quanto aos procedimentos adotados para alunos em risco de evadir. Tais especificidades são apresentadas a seguir.

#### 2.1.1 Procedimentos da UnB Para Alunos com Risco de Evadir

Os critérios para que um aluno seja desligado na UnB são apresentados na Figura 2.1.

A abordagem da UnB para evitar desligamento consiste de separar os alunos em dois grupos (alunos cumprindo condição para não serem desligados e alunos que não estão em

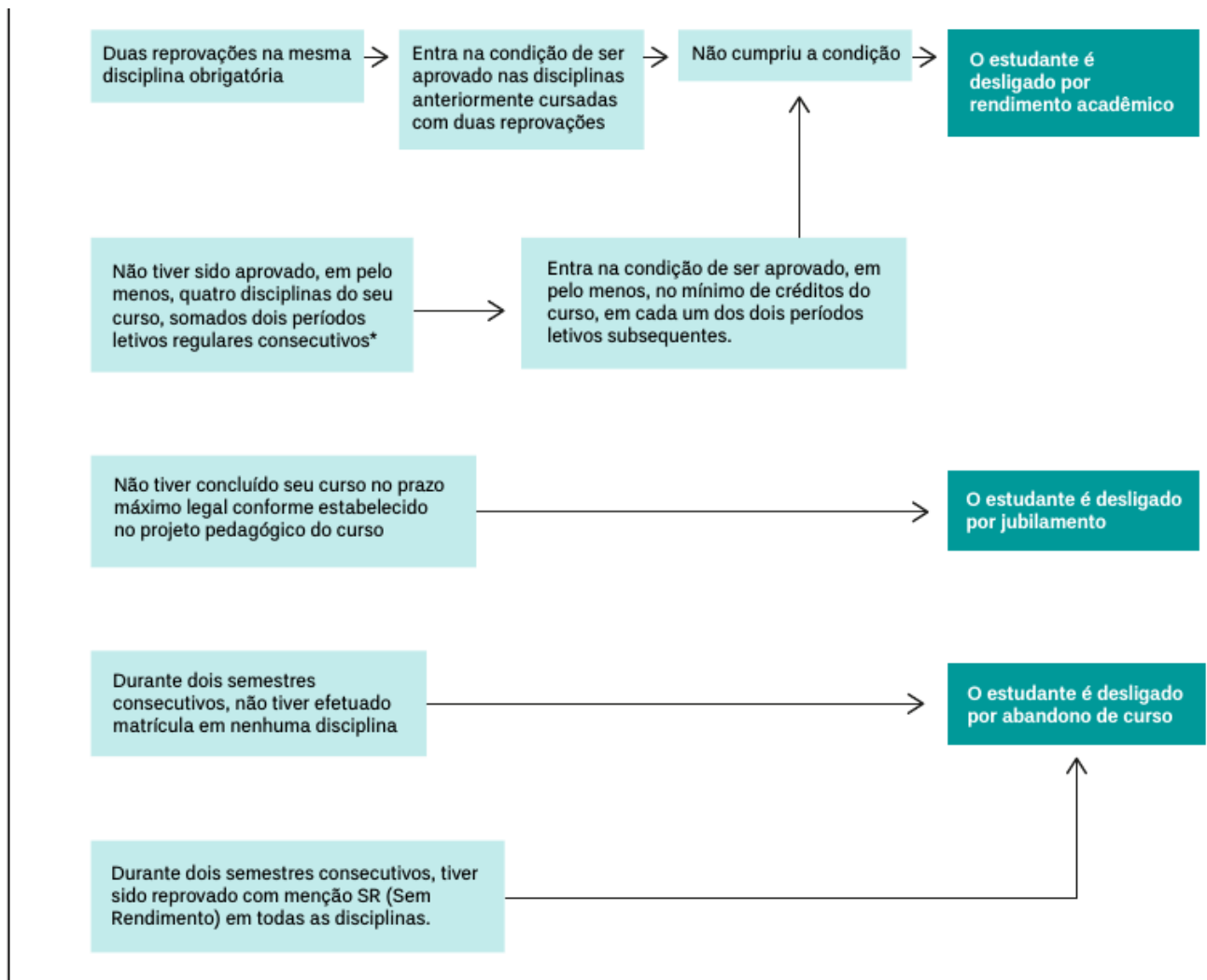


Figura 2.1: Critérios Para Desligamento na UnB

condição) e ter os alunos em condição supervisionados por um orientador. Os critérios para que um aluno esteja em condição são [6]:

- ter duas reprovações na mesma disciplina obrigatória.
- não ser aprovado em quatro disciplinas do curso em dois períodos regulares consecutivos.
- chegar ao último período letivo do curso, segundo o prazo máximo legal, sem a possibilidade de concluí-lo.

### 2.1.2 O Índice de Rendimento Acadêmico

Outra particularidade da UnB diz respeito às notas que seus alunos recebem. Para cada matéria, os alunos podem receber as menções apresentadas na Tabela 2.1 [6]:



Tabela 2.1: Equivalência entre Menção Obtida por um Aluno e Nota Final na Disciplina

Menção	Nota de 0 a 10
SS	9,0 a 10
MS	7,0 a 8,9
MM	5,0 a 6,9
MI	3,0 a 4,9
II	0,1 a 2,9
SR	0

Cada aluno tem também um índice de rendimento acadêmico (IRA), um valor entre 0 e 5 que sumariza as menções obtidas por um aluno, desde que entrou na UnB. Segundo [6], O IRA é calculado da seguinte forma:

$$IRA = (1 - \frac{0,6 * DT_b + 0,4 * DT_p}{DC}) * (\frac{\sum_i P_i * CR_i * Pe_i}{\sum_i CR_i * Pe_i}) \quad (2.1)$$

em que os símbolos da fórmula são explicados a seguir:

- $DT_b$ : Número de disciplinas obrigatórias trancadas
- $DT_p$ : Número de disciplinas optativas trancadas
- $DC$ : Número de disciplinas matriculadas (incluindo as trancadas)
- $P_i$ : Peso da menção, onde (SS = 5, MS = 4, MM = 3, MI = 2, II = 1, SR = 0)
- $Pe_i$ : Período em que uma dada disciplina foi cursada, obedecendo à limitação min(6, Período). O período corresponde ao semestre em que o aluno está em seu curso.
- $CR_i$ : Número de créditos de uma disciplina

O objetivo do IRA é fornecer um valor que sumariza o desempenho acadêmico de um aluno durante todo o período em que este esteve na UnB. Nessa pesquisa, o IRA é utilizado para o cálculo dos atributos de desempenho acadêmico: coeficiente de melhora acadêmica, média do período e posição em relação ao semestre que ingressou.

## 2.2 CRISP-DM

O CRISP-DM(do inglês *CRoss-Industry Standard Process for Data Mining*) é um modelo de referência para mineração de dados que se divide em seis fases principais [7]:

1. Entendimento do negócio

2. Entendimento dos dados
3. Preparação dos dados
4. Modelagem
5. Avaliação
6. Implantação

Cada uma dessas fases é explicada a seguir. Essa pesquisa cobre as cinco primeiras fases do CRISP-DM.

### 2.2.1 Entendimento do Negócio

Nesta etapa determinam-se os objetivos do negócio (através do entendimento do que o cliente almeja), avalia-se a situação atual, determinam-se os objetivos da mineração de dados e produz-se um planejamento para realização do projeto [7]. Uma importante tarefa dentro dessa fase do CRISP-DM é o levantamento das práticas atuais na área.

### 2.2.2 Entendimento dos Dados

Essa fase é composta pelas tarefas de: coleta inicial dos dados, descrição dos dados, exploração dos dados e verificação da qualidade dos dados [7]. Na fase de entendimento dos dados, juntamente com a fase de preparação dos dados, é comum a utilização de estatística descritiva. A estatística descritiva permite identificar problemas de qualidade nos dados como, por exemplo, a identificação de valores de atributos faltantes ou a identificação de *outliers* ou a identificação de atributos com cardinalidade irregular [8]. Técnicas comuns de estatística descritiva incluem histogramas, gráficos de barra e *boxplots*.

### 2.2.3 Preparação dos Dados

Essa fase é composta pelas tarefas de: seleção dos atributos, limpeza dos dados, enriquecimento dos dados, integração dos dados e formatação dos dados [7]. Para a tarefa de seleção dos dados, é útil ter uma noção do grau de correlação entre as variáveis. Caso a correlação entre algum par de variável seja muito alta, a eliminação de uma delas pode acarretar em um modelo mais simples de ser compreendido (sem prejudicar o desempenho).

Assim, pode-se usar um coeficiente de correlação para medir o grau de dependência entre atributos numéricos e decidir pela eventual eliminação de algum. Existem vários testes para se medir a correlação entre variáveis, como o coeficiente de Spearman e o coeficiente de correlação de Kendall [9].

### 2.2.4 Modelagem

Essa fase é composta pelas tarefas de: seleção da técnica de modelagem, geração dos casos de teste, construção do modelo e avaliação do modelo [7]. É nessa fase que utilizam-se os algoritmos de aprendizagem de máquina como árvores de decisão, redes neurais e SVMs [10]. Nessa fase ocorre, comumente, a divisão dos dados em treino, validação e teste. Um cuidado comum a ser tomado nessa fase é a redução do *overfitting*. Métricas comuns para avaliar o desempenho dos modelos incluem a precisão, o *recall* e a *F-measure*.

### 2.2.5 Avaliação

Enquanto a tarefa de avaliação da fase de modelagem tem como objetivo lidar com fatores como a precisão e a generalidade do modelo, essa fase tem como objetivo averiguar o quanto o modelo cumpre com os objetivos de negócio [7]. São tarefas dessa fase: avaliação de resultados, revisão de processos e decisão de quais serão os próximos passos.

### 2.2.6 Implantação

A fase final do CRISP-DM é a implantação. Nessa fase, são realizadas as tarefas de: planejar a implantação, planejar monitoramento e manutenção, produzir relatório final e revisar o projeto [7].

## 2.3 Algoritmos de Aprendizagem de Máquina

Nesta seção fazem-se breves explicações sobre os algoritmos de aprendizagem de máquina usadas na pesquisa. Explica-se sobre as árvores de decisão e *random forests*, regressor linear, SVRs, redes neurais, *Naive Bayes* e o ZeroR. Algoritmos de aprendizagem de máquina são utilizados para a indução de modelos em tarefas de mineração de dados.

As principais tarefas de aprendizagem de máquina são classificação e regressão [10]. Na classificação, a variável de saída é classificada em valores discretos, enquanto que na regressão a variável de saída assume valores contínuos.

Outra questão importante em aprendizagem de máquina diz respeito à minimização de *overfitting*. *Overfitting* é o fenômeno no qual conseguir um ajuste aos fatos observados (dados) bem não mais garante que teremos um bom desempenho fora da amostra, podendo ocorrer, na verdade, o efeito oposto [10].

### 2.3.1 Árvores de Decisão e Random Forest

Árvore de Decisão é um algoritmo de aprendizagem de máquina bastante usada em pesquisas científicas, em uma variedade de contextos [11]. Tal algoritmo pode ser usado tanto para classificação quanto para regressão. O modelo induzido adota uma representação em árvore, com cada vértice fazendo referência à um teste a ser feito para um atributo de uma instância e cada uma das arestas indicando um dos possíveis valores do atributo. A Figura 2.2, exemplo encontrado em [11] mostra uma árvore de decisão para o problema de decidir se uma manhã de sol está adequada para a prática de jogar tênis. Note que as folhas das árvores correspondem à decisão à ser tomada.

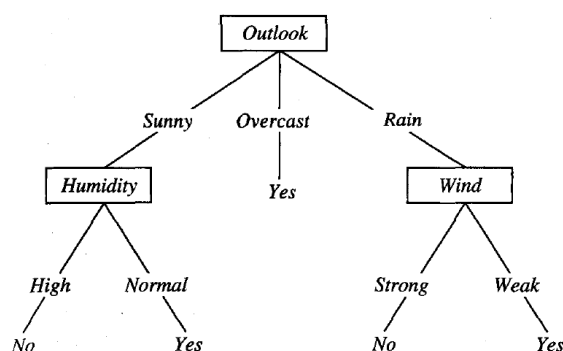


Figura 2.2: Árvore de Decisão Para a Prática de Tênis

Por fim, *Random Forests* é um algoritmo de aprendizagem de máquina (baseado nas árvores de decisão) adequado para as tarefas de classificação e regressão. *Random Forests* operam construindo várias árvores de decisão durante o treino e fornecendo como saída a predição média (no caso de regressão) ou a moda das classes (no caso de classificação) [12].

### 2.3.2 Regressão Linear

O regressor linear é uma abordagem linear, utilizada na tarefa de regressão, que tenta prever a relação entre uma variável dependente  $y$  em termos de uma ou mais variáveis independentes, denotadas por  $X$ . Se apenas uma variável independente é usada, a regressão linear é dita simples, enquanto que se mais de uma variável é utilizado, a regressão linear é dita múltipla.

Por fim, pode-se tentar prever múltiplas variáveis dependentes correlacionadas, caso no qual o modelo é dito regressor linear multivariado. Nessa pesquisa, fez-se regressão linear múltipla multivariada. De modo geral, modelos lineares costumam não ser muito propensos à *overfitting* e são boas alternativas iniciais para problemas de aprendizagem de máquina [8].

### 2.3.3 SVR

SVM (do inglês *Support Vector Machine*) são um conjunto de métodos de aprendizagem de máquina que podem ser usados para classificação, regressão e detecção de *outliers*. Por meio de seu *kernel*, SVMs conseguem realizar transformações não lineares de alta dimensão [10] e assim resolver problemas de domínios diferentes. As Figuras ?? e ?? mostram SVM sendo usadas para tarefas de classificação com dois tipos de kernel diferentes: linear e RBF.

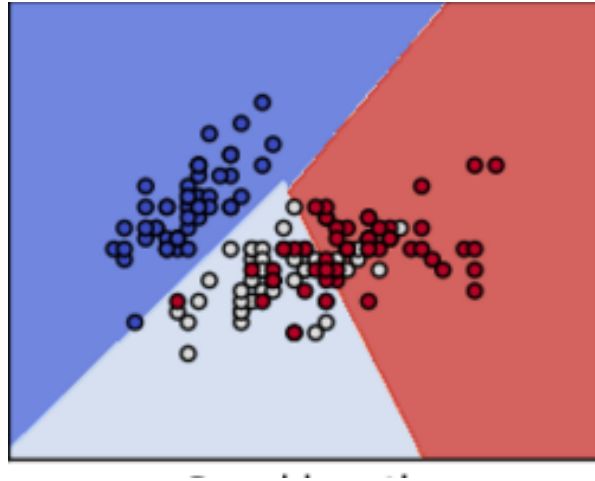


Figura 2.3: Exemplo de SVM com Kernel Linear

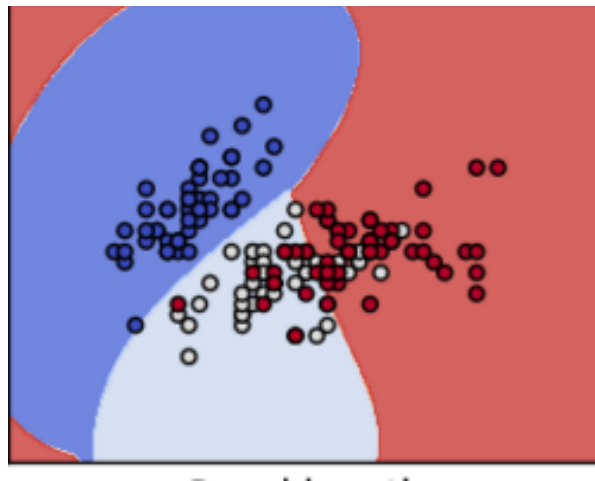


Figura 2.4: Exemplo de SVM com Kernel RBF

Tal algoritmo de aprendizagem de máquina é fácil de usar e tem bom desempenho na prática, o que explica assim sua popularidade [10]. O método utilizado na SVM pode ser estendido para regressão. Nesse caso, o método é chamado SVR (do inglês *Support Vector Machine for Regression*).

### 2.3.4 ANN

ANN (do inglês *Artificial Neural Networks*) ou redes neurais são uma técnica de aprendizagem de máquina inspirado na organização dos neurônios que tem bastante sucesso em aplicações de áreas diversas, indo de visão computacional [10] à estratégia de jogos como Go [13]. Entretanto, como redes neurais podem induzir modelos muito flexíveis e com grande poder de aproximação, é fácil incorrer em *overfitting* ao se escolher tal técnica [10]. ANN podem ser usadas tanto para classificação quanto para regressão.

ANN são um conjunto de unidades conectadas, chamadas de neurônios (em analogia aos neurônios do cérebro humano). Se dois neurônios estão conectados (essa conexão é dita sinapse), um deles pode transmitir um sinal ao outro. As sinapses tem um peso, que varia conforme ocorre o aprendizado, e pondera o sinal propagado.

Os neurônios tipicamente são organizados em camadas, as quais realizam diferentes transformações nas entradas que recebem. Costuma-se diferenciar entre a camada de entrada, a camada escondida e a camada de saída, conforme pode ser visto na Figura 2.5. Assim, o funcionamento de uma rede neural ocorre, de modo bem superficial, da seguinte forma: uma nova instância é fornecida à rede neural (na camada de entrada), ativam-se determinados neurônios da rede neural (de acordo com os dados de entrada) e um determinado valor é fornecido na camada de saída, de acordo com os neurônios que foram ativados.

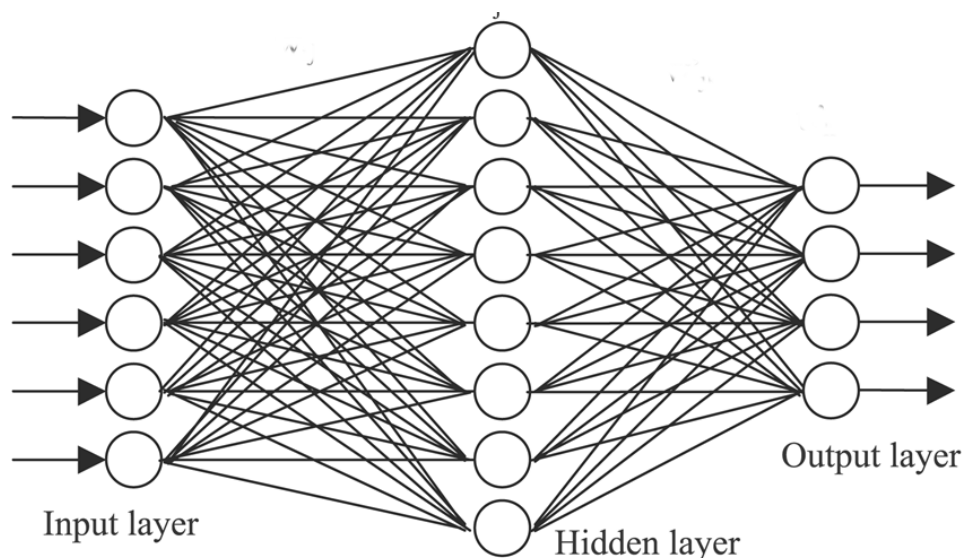


Figura 2.5: Camadas de uma Rede Neural

### 2.3.5 Naive Bayes

O *Naive Bayes* é um classificador probabilístico baseado na aplicação do teorema de Bayes com a premissa de independência entre os atributos [8]. Um parâmetro importante para tal algoritmo é o tipo de distribuição que se assume que os atributos possuem. Alguns exemplos de distribuições são: Gaussiana, multinomial e Bernoulli.

Seja  $y$  a variável de saída e seja a entrada um vetor de atributos, de  $x_1$  até  $x_n$ . A hipótese de independência entre atributos é descrita pela Fórmula 2.2. Por fim, o teorema de Bayes com a hipótese de independência é dado pela Fórmula 2.3.

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y) \quad (2.2)$$

$$P(y|x_1, \dots, x_n) = \frac{P(y) \prod_{1 \leq i \leq n} P(x_i|y)}{P(x_1, \dots, x_n)} \quad (2.3)$$

A Figura ?? ilustra como o Naive Bayes funciona. O classificador probabilístico, representado por  $C$ , decide com base nos atributos à ele passados (representados por  $X_1, X_2, X_3, X_4$ ). Na Figura ?? não há relação entre  $X_1, X_2, X_3$  e  $X_4$ ; representando o fato de que os atributos são considerados condicionalmente independentes.

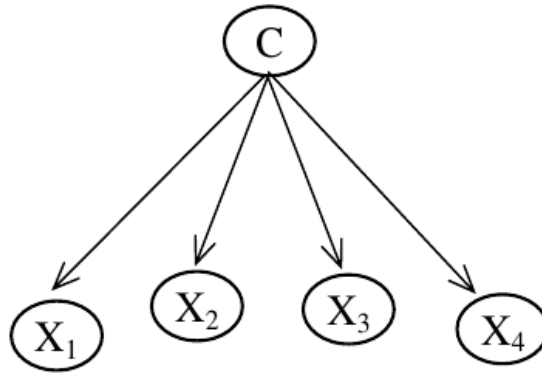


Figura 2.6: Diagrama do Classificador Probabilístico Naive Bayes

### 2.3.6 ZeroR

O ZeroR é um dos métodos de classificação mais simples. Tal método ignora os atributos e sempre prevê a classe majoritária. Embora não tenha um grande poder de predição, o ZeroR é útil para determinar um desempenho básico, que os algoritmos de aprendizagem de máquina devem ser capazes de superar.

## 2.4 Métricas Para Avaliação de Desempenho dos Modelos

Existem várias métricas para a avaliação do desempenho de modelos induzidos. Dentre as técnicas comuns para o problema de classificação supervisionado, pode-se citar a precisão, o *recall* e a *F-measure*.

A precisão é definida como sendo a razão entre a quantidade de verdadeiros positivos ( $TP$ ) e a quantidade de positivos ( $TP + FP$ ).  $TP$  significa verdadeiro positivos e  $FP$  significa falso positivo. Assim sendo, a precisão traduz o quão frequentemente o modelo acerta quando este faz uma previsão positiva. A fórmula para o cálculo da precisão é mostrada a seguir:

$$Precisão = \frac{TP}{TP + FP} \quad (2.4)$$

Já o *recall* é definido como a razão entre a quantidade de verdadeiros positivos pela soma de verdadeiros positivos com os falsos negativos. Desse modo, *recall* traduz o quão confiante podemos estar que todas as instâncias positivas foram encontradas pelo modelo. A fórmula para o cálculo do *recall* é mostrada a seguir:

$$Recall = \frac{TP}{TP + FN} \quad (2.5)$$

Precisão e *recall* fornecem diferentes informações, ambas úteis. Desse modo, faz sentido definir uma métrica que leve em conta ambas as informações para avaliar o desempenho. A *F-measure* é essa métrica, sendo definida como a média harmônica ponderada entre a precisão e o *recall*. Precisão, *recall* e *F-measure* assumem valores de 0 a 1. A Fórmula 2.6 é geral. Note que a ponderação depende de um parâmetro  $\beta$ . Para o caso de precisão e *recall* terem ponderações iguais, o cálculo se reduz à Fórmula 2.7.

$$F - measure = (1 + \beta^2) * \frac{precisão * recall}{\beta^2 * precisão + recall} \quad (2.6)$$

$$F - measure = 2 * \frac{precisão * recall}{precisão + recall} \quad (2.7)$$



# Capítulo 3

## Metodologia

Neste capítulo, descreve-se a metodologia usada em toda a pesquisa. Detalha-se como foi feito o levantamento do estado da arte, a obtenção e a utilização dos dados, a seleção de atributos, a eliminação de atributos devido à *missing values*, a eliminação de *outliers*, a análise preliminar por meio de estatística descritiva, a eliminação de atributos relacionados ou irrelevantes, as divisões em treino e teste e as divisões em semestre, a escolha dos algoritmos de aprendizagem de máquina, e a forma de avaliar o desempenho destes.

Faz-se nesse parágrafo um breve comentário acerca da metodologia usada. A metodologia de mineração de dados usada é baseada no modelo CRISP-DM. Foram utilizados dados reais descaracterizados de alunos da UnB e de seus desempenhos. Para induzir os modelos e avaliar o desempenho deles, os dados foram divididos em dados de treino, validação e teste, segundo a abordagem *holdout*. Os dados dos cursos analisados foram agrupados nos grupos: licenciatura, FT e computação. Os modelos preditos foram induzidos com os algoritmos Naive Bayes, Random Forest, ANN, regressor linear e SVR, disponibilizados pela biblioteca `scikit-learn` (v. 0.18.1) da linguagem Python. A performance dos modelos foi mensurada com a métrica *F-measure*, e comparada à obtida com o classificador ZeroR.

### 3.1 Levantamento do Estado da Arte

Foi feito o levantamento do estado da arte através da leitura de diversos artigos. Procurou-se assim compreender quais fatores são importantes para a evasão [14] [1] [15], como técnicas de aprendizagem de máquina podem ser utilizadas para resolver o problema [14] [16] e como trabalhar especificamente com os dados da UnB [6] [2]. Explica-se brevemente a seguir os artigos citados.

O excelente artigo [14] foi o mais útil para a pesquisa em questão. Neste artigo, utiliza-se mineração de dados para avaliar o risco de um aluno ficar na universidade mais tempo

que o previsto. Os dados são da Universidade Federal de Pernambuco. De modo bastante interessante, avalia-se também a viabilidade econômica da implementação de um processo de aconselhamento.

[15] e [1] analisam alguns fatores que podem vir a ser importantes para a evasão. Em [15], os autores investigam razões para se abandonar a matéria introdutória de ciência da computação na universidade de Helsinki. Embora vários motivos tenham sido elencados, ressalta-se a falta de tempo e de motivação dos estudantes. Já em [1], estuda-se e constata-se que cursos que requerem maior abstração algorítmica e conhecimento matemático tem índices de evasão superiores. Além disso, em [1], constata-se que a relação candidato por vaga é inversamente proporcional à evasão.

No artigo [16], utiliza-se mineração de dados para identificar quais alunos tem maiores chances de retenção. As técnicas usadas incluem, árvores de classificação, ANNs e MARS.

No artigo [2], aplicam-se técnicas de análise de sobrevivência para estudantes de computação da Universidade de Brasília (assim como a pesquisa descrita nesta monografia). Os dados são de 2005 até 2015.

Por fim, a pesquisa aqui descrita pode também ser vista dentro da área de *Customer Churn*. Nesse caso, a UnB seria considerada a empresa e seus alunos seriam considerados seus clientes. A pesquisa usa aprendizagem de máquina para identificar clientes (alunos) com maior risco de romperem com a empresa (UnB). Nesse caso, ao conceito de rompimento corresponderiam evasão ou migração. No artigo [17], aprendizagem de máquina é usada na área de *Customer Churn*.

## 3.2 Obtenção e Utilização dos Dados

Obtiveram-se informações descaracterizadas relativas aos dados sociais e ao desempenho acadêmico de alunos de graduação da UnB obtidos do SIGRA - Sistema de Informação da Graduação. Todos os dados utilizados vieram de uma só fonte, de modo que o comum problema encontrado na área de mineração de dados de garantir a consistência dos dados entre várias fontes não foi enfrentado.

Optou-se por restringir a pesquisa apenas aos alunos que entraram a partir de 2000 e saíram até 2016, de modo a trabalhar com dados mais recentes. Para simplificar a análise, decidiu-se trabalhar apenas com uma área específica (computação), de modo que apenas os seguintes cursos foram considerados:

- Ciência da Computação (Bacharelado)
- Computação (Licenciatura)
- Engenharia da Computação

- Engenharia de Redes
- Engenharia de Software
- Engenharia Mecatrônica

### 3.3 Seleção Preliminar de Atributos

Com base no levantamento do estado da arte feito, selecionou-se quais atributos teriam melhor condição de serem significativos para que um aluno fosse ou não desligado. Assim sendo, lista-se a seguir os atributos sociais considerados em uma análise inicial:

- Cotista (ou não)
- Curso
- Forma de Ingresso
- Idade
- Raça
- Sexo
- Tipo da Escola

Além de dados sociais, utilizaram-se os seguintes atributos (relativos ao desempenho acadêmico):

- Coeficiente de Melhora Acadêmica
- Indicador de Aluno em Condição
- Média do Período
- Posição em relação ao semestre que ingressou
- Quantidade de créditos já integralizados
- Taxa de Aprovação, Taxa de Reprovação e Taxa de Trancamento
- Taxa de aprovação na disciplina mais difícil de cada semestre

O coeficiente de melhora acadêmica é definido como sendo a razão entre o IRA do aluno em um semestre pelo IRA do aluno no semestre anterior. Dessa maneira, o coeficiente de melhora acadêmica mostra se o desempenho do aluno está melhorando, piorando ou se encontra estável. Nesse atributo, apenas as notas de um aluno em um semestre são consideradas: não se mantém um histórico de todos os semestres.

A média do período é calculada da mesma forma que o IRA, mas considerando apenas as menções obtidas em um determinado semestre. Assim como o IRA, a média do período varia entre 0 e 5.

A posição em relação ao semestre  $P$  para um determinado aluno é definida como sendo: o número de alunos com IRA maior que o do estudante em questão (considerando apenas aqueles que entraram no mesmo curso do estudante, no mesmo ano e no mesmo semestre). Assim, um aluno com posição  $P = 0$  é aquele que tem o maior IRA em relação a seus colegas que entraram no mesmo curso durante o mesmo ano e semestre.

Como todos os cursos da UnB requerem uma quantidade de créditos mínima para graduação, incluiu-se o atributo quantidade de créditos já integralizados.

A taxa de aprovação é definida como a razão entre o número de matérias cursadas pelo aluno com aprovação pelo número de matérias cursadas pelo aluno. Analogamente, a taxa de reprovação é definida como a razão entre o número de matérias cursadas pelo aluno com reprovação pelo número de matérias cursadas pelo aluno. Deve-se dizer que incluir a taxa de aprovação e a de reprovação não é, a priori, redundante, já que além de ser aprovado ou reprovado em uma matéria, outra possibilidade é o aluno realizar o trancamento. Pensando nisso, definiu-se a taxa de trancamento como sendo a razão entre o número de matérias trancadas pelo aluno pelo número de matérias cursadas pelo aluno.

Por fim, a taxa de aprovação na disciplina mais difícil do semestre é definida como sendo a razão entre o número de aprovações na disciplina mais difícil do semestre pelo número de semestres na UnB. A disciplina considerada como a mais difícil do semestre é aquela com a maior taxa de reprovação, independente dela ser obrigatória ou não.

Deve-se destacar que, para um mesmo aluno, os atributos relacionados ao desempenho variam conforme o semestre considerado. O mesmo não ocorre para os atributos sociais.

### 3.4 Eliminação de Atributos Devido a Missing Values

Optou-se por eliminar atributos cuja quantidade de entradas com *missing values* fosse superior à 40%. Assim, eliminaram-se os atributos raça e tipo da escola.

A Figura 3.1 mostra o gráfico de barra para o atributo raça, enquanto que a Figura 3.2 mostra o gráfico de barra para o atributo tipo da escola.

### 3.5 Eliminação de Outliers

Decidiu-se não trabalhar com casos de alunos que após ingressar na universidade não demonstraram interesse em cursar matérias (por exemplo, aqueles que reprovaram em todas as disciplinas com SR). Tais casos foram tratados como *outliers*. Após a análise

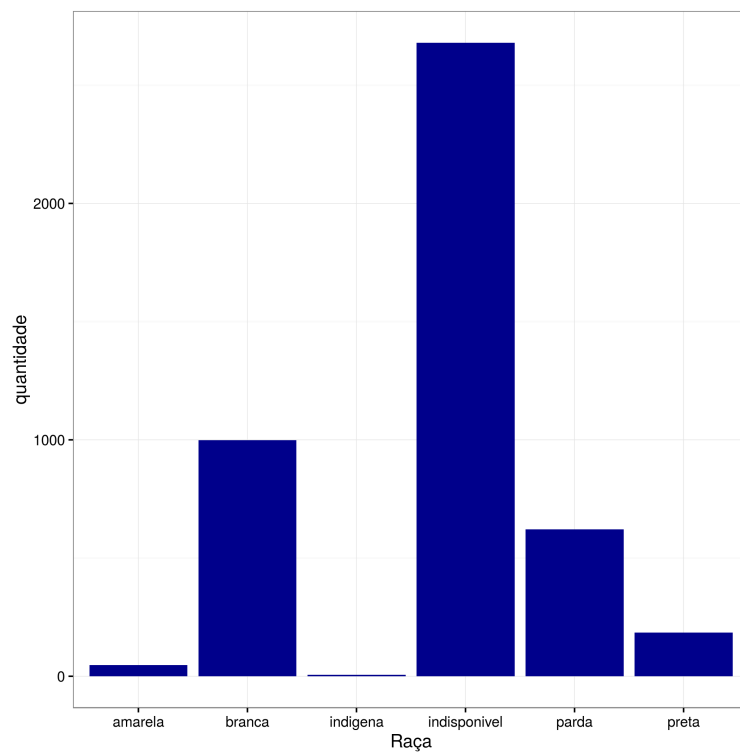


Figura 3.1: Gráfico de Barra para Atributo Raça

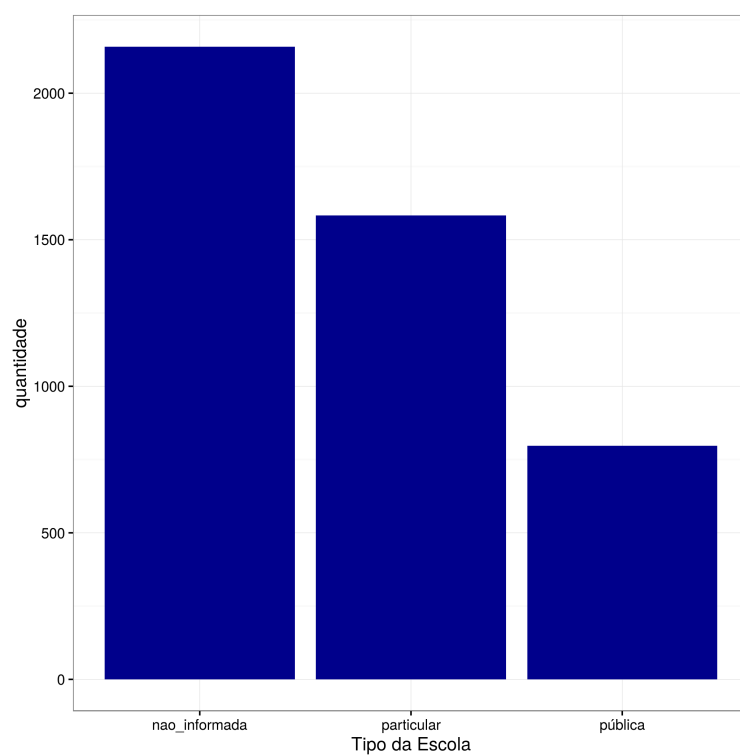


Figura 3.2: Gráfico de Barra para Atributo Tipo da Escola

individual de cada caso, os *outliers* foram eliminados do espaço amostral. Eliminaram-se 233 estudantes do espaço amostral dessa maneira, ficando-se assim com 4536 estudantes.

## 3.6 Estatística Descritiva

Foi feita uma análise preliminar por meio de estatística descritiva. As subseções seguintes explicam cada um dos procedimentos adotados: a mudança na base de dados realizada, a mudança nos valores de alguns atributos e os gráficos de barra e histogramas finais.

### 3.6.1 Mudança na Base de Dados

Foi possível observar que os atributos variavam significativamente de acordo com o curso. Isso se deve ao fato de cada curso ter currículo diferente dos demais, alguns cursos serem de instituições diferentes (Faculdade de Tecnologia (FT) ou Instituto de Ciências Exatas (IE), por exemplo) e os cursos possuírem “maturidade” diferentes (devido à data início de cada curso ser diferente). Outra observação preliminar possível foi a de que a proporção de alunos que ingressam com idade avançada que se forma é menor do que a de alunos mais jovens. Após a aplicação da técnica de tabela de contingência, essas observações levaram a partição da base de dados original em quatro bases de dados:

- Alunos Jovens da FT: contém todos os alunos que ingressaram com 30 anos ou menos que cursam Engenharia de Redes ou Engenharia Mecatrônica. Tais cursos tem a peculiaridade de estarem associados à FT, diferentemente de todos os demais.
- Alunos Jovens de Licenciatura: contém todos os alunos que ingressaram com 30 anos ou menos que cursam Computação (Licenciatura). O curso de Licenciatura tem a peculiaridade de ser o único noturno.
- Alunos Jovens de Computação: contém todos os alunos que ingressaram com 30 anos ou menos que cursam Ciência da Computação, Engenharia da Computação ou Engenharia de Software.
- Alunos Seniores: contém todos os alunos com mais de 30 anos.

Devido à baixa quantidade de alunos seniores, não se separou tal categoria em grupos.

A evidência fornecida pela estatística descritiva é mostrada a seguir. A Tabela 3.1 mostra a percentagem de alunos capazes de formar de acordo com o curso, enquanto que a Tabela 3.2 mostra a percentagem de alunos capazes de formar de acordo com a idade, separando aqueles que entram na UnB com até 30 anos daqueles que entram na UnB com mais de 30 anos.

Tabela 3.1: Percentagem de Alunos Capazes de Formar por Curso

Curso	Percentagem capaz de Formar
Ciência da Computação (Bacharelado)	42%
Computação (Licenciatura)	27%
Engenharia da Computação	20%
Engenharia de Redes	48%
Engenharia de Software	49%
Engenharia Mecatrônica	47%

Tabela 3.2: Percentagem de Alunos Capazes de Formar por Idade

Idade	Percentagem capaz de Formar
Estudantes que entraram com até 30 anos	41%
Estudantes que entraram com mais de 30 anos	11%

### 3.6.2 Mudança nos Valores de Atributos

Dois atributos tiveram alguns de seus valores agrupados em categorias para facilitar o posterior tratamento dos dados. Tais atributos são a forma de entrada e a forma de saída. A distribuição original de tais atributos é mostrada a seguir.

O gráfico de barra para o atributo forma de ingresso, com seus valores originais, é mostrado na Figura 3.3. Por questões de legibilidade, a legenda no gráfico foi encurtada. Seu significado é apresentado a seguir:

- **vest:** Ingresso via Vestibular
- **ci:** Ingresso via Convênio (Intercâmbio)
- **to:** Ingresso via Transferência Obrigatória
- **ac:** Ingresso via Acordo Cultural PEC
- **ca:** Ingresso via Convênio Andifes
- **mc:** Ingresso via Matrícula Cortesia
- **tf:** Ingresso via Transferência Facultativa
- **ppp:** Ingresso via PEC-G Peppfol (Programa de Pesquisa e Ensino de Português para Falantes de Outras Linguagens)
- **pdcs:** Ingresso via Portador de Diploma de Curso Superior

- **vmc**: Ingresso via Vestibular para Mesmo Curso

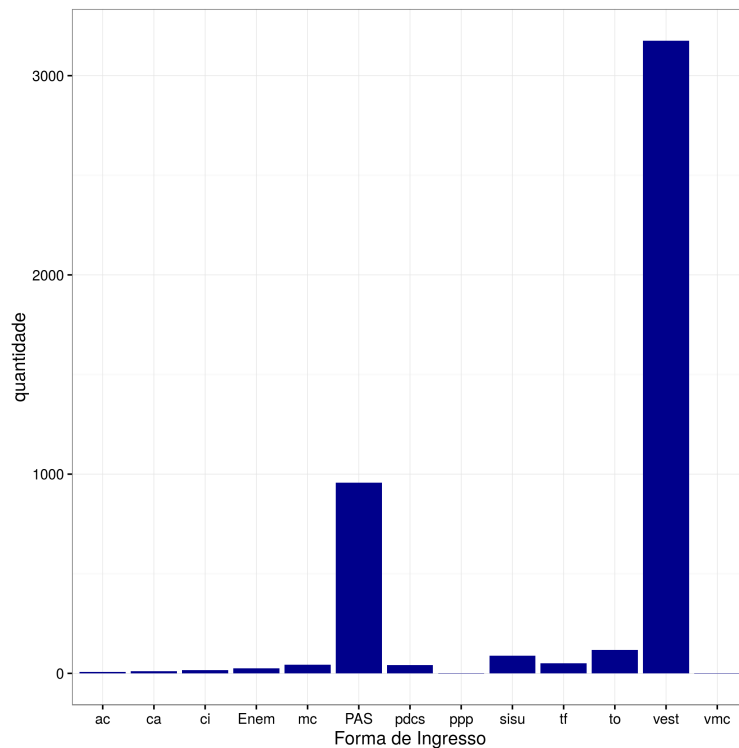


Figura 3.3: Gráfico de Barra de Todos os Alunos, para o Atributo Forma de Ingresso

O gráfico de barra para o atributo forma de saída, com seus valores originais, é mostrado na Figura 3.4. Por questões de legibilidade, a legenda no gráfico foi encurtada. Seu significado é apresentado a seguir:

- **deslg**: Saída por desligamento
- **form**: Saída pois conseguiu formar
- **trnsf**: Saída devido à Transferência
- **vest**: Saída devido à novo vestibular

No atributo forma de entrada, de modo a eliminar os vários valores com poucas instâncias na base de dados, criou-se a categoria “outros”. Assim, apenas três categorias foram consideradas: vestibular, PAS e outros.

Decidiu-se trabalhar com apenas três valores de forma de saída: graduou, evadiu e migrou. Essa última categoria foi criada para agrupar os casos de transferência, mudança de curso, mudança de turno e realização de um novo vestibular.



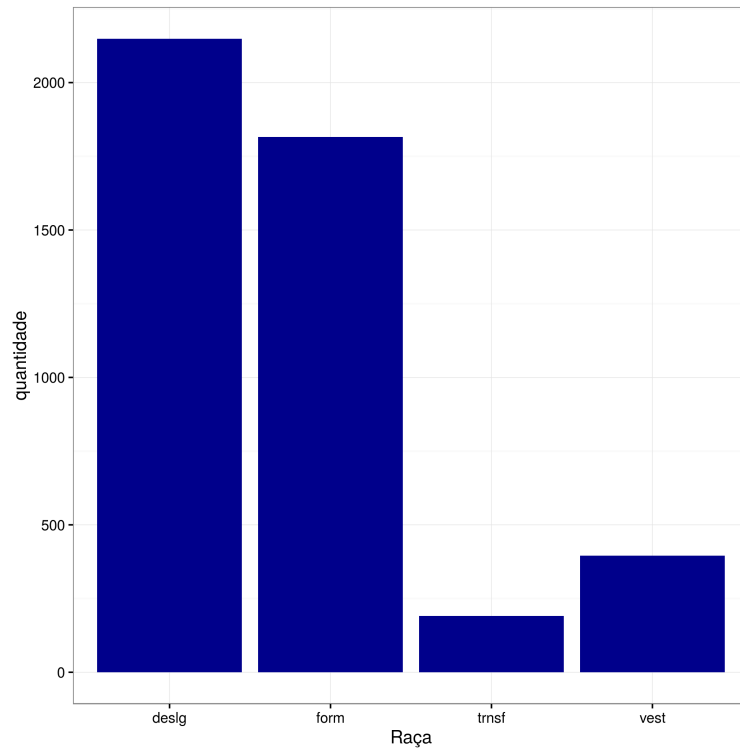
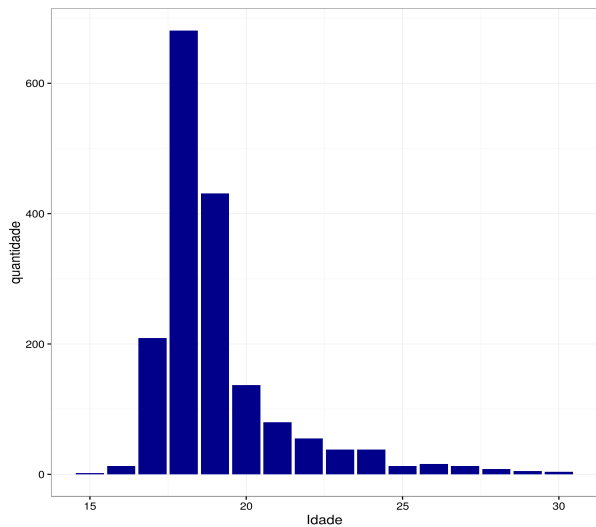


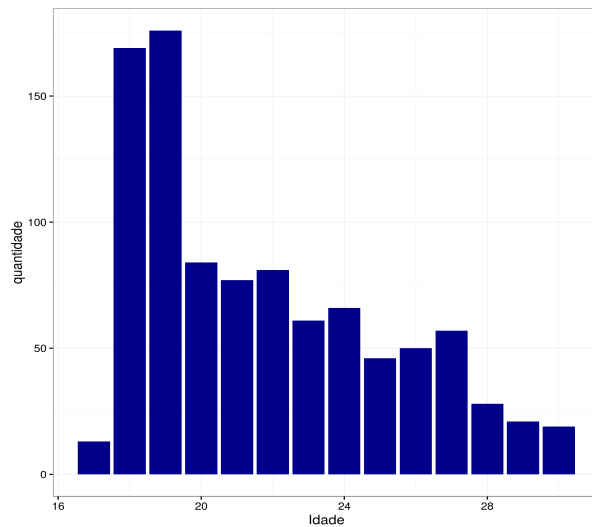
Figura 3.4: Gráfico de Barra para Atributo Forma de Saída

### 3.6.3 Gráficos de Barra e Histogramas

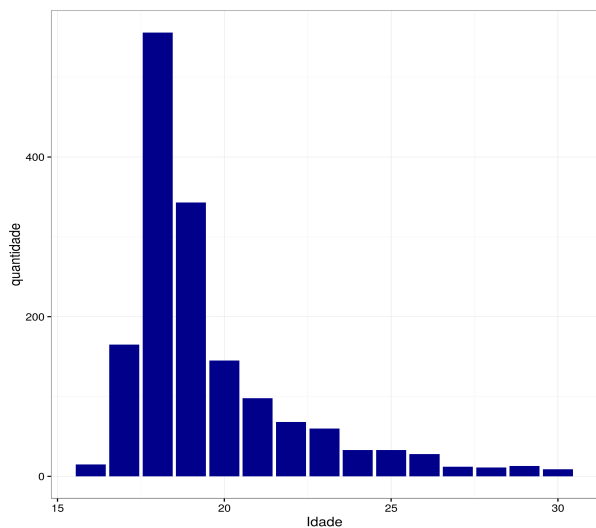
Após as modificações descritas nas subseções anteriores, fez-se o gráfico de barra para os atributos discretos e os histogramas para os atributos contínuos. Considerou-se, para isso, a divisão dos dados nas quatro bases de dados descritas. Mostra-se a seguir a distribuição dos atributos por meio de gráficos de barra e histogramas. Essa parte de estatística descritiva levou em consideração a divisão dos dados nas 4 bases de dados. Para se ter uma ideia de como uma determinada base de dados se comportou em relação ao conjunto de todos os dados, fez-se também gráficos de barra e histogramas considerando todos os alunos das 4 bases de dados.



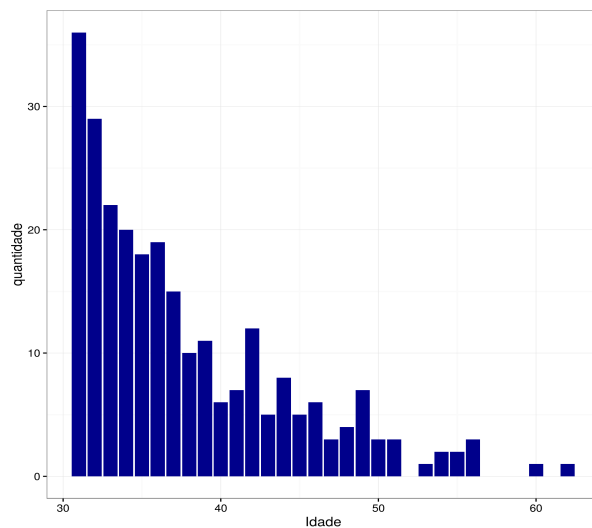
(a) Alunos Jovens da FT



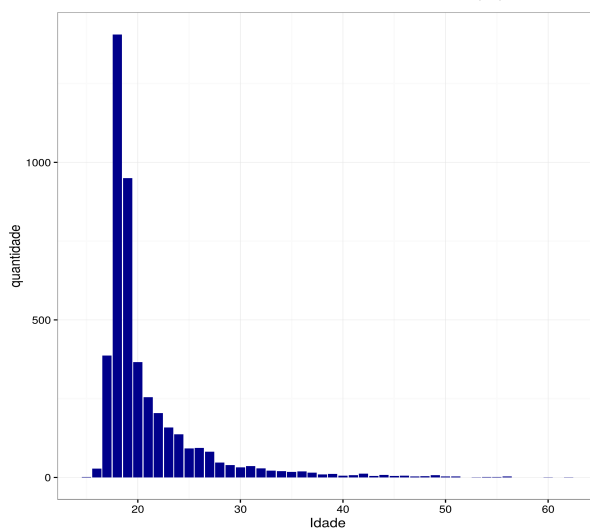
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação



(d) Alunos Seniores



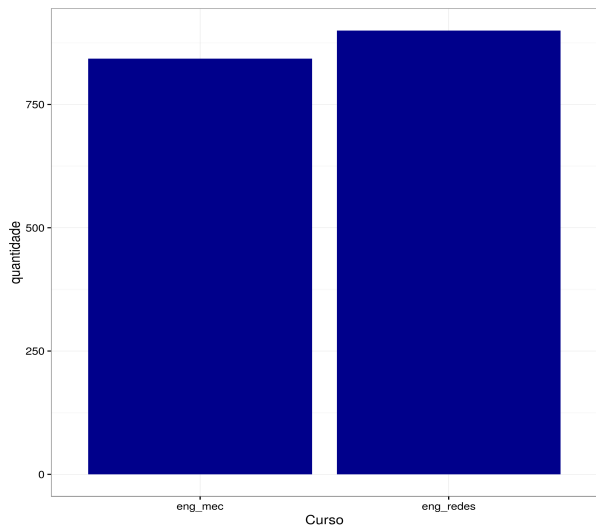
(e) Todos os Alunos

Figura 3.5: Atributo Idade, Conforme os Diferentes Modelos

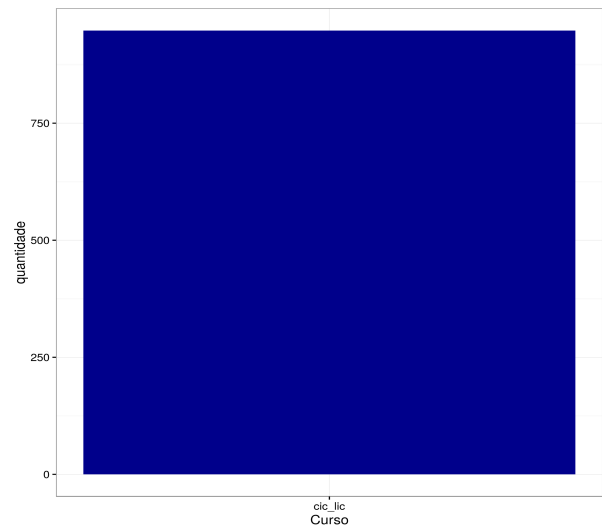
Por questões de legibilidade, a legenda na Figura 3.6 foi encurtada. Seu significado é apresentado a seguir:

- `cic_b`: Alunos de Ciência da Computação (Bacharelado).
- `cic_lic`: Alunos de Computação (Licenciatura).
- `eng_comp`: Alunos de Engenharia da Computação.
- `eng_mec`: Alunos de Engenharia Mecatrônica.
- `eng_redes`: Alunos de Engenharia de Redes.
- `eng_softw`: Alunos de Engenharia de Software

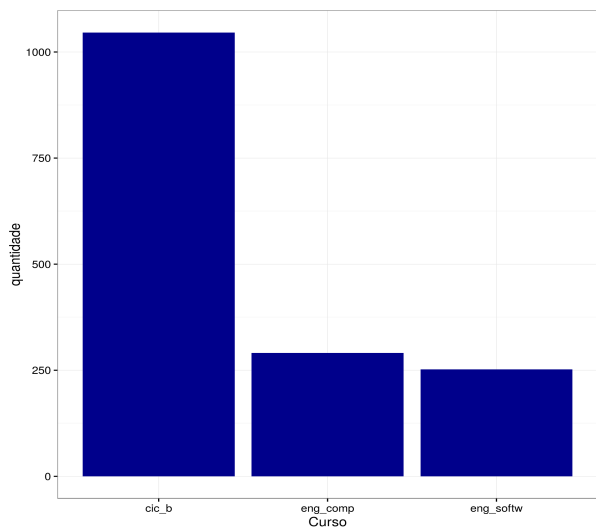
No imagem correspondente aos alunos jovens da licenciatura, o gráfico aparece de modo um pouco estranho. Isso ocorre porque há apenas um curso nessa categoria: licenciatura em computação.



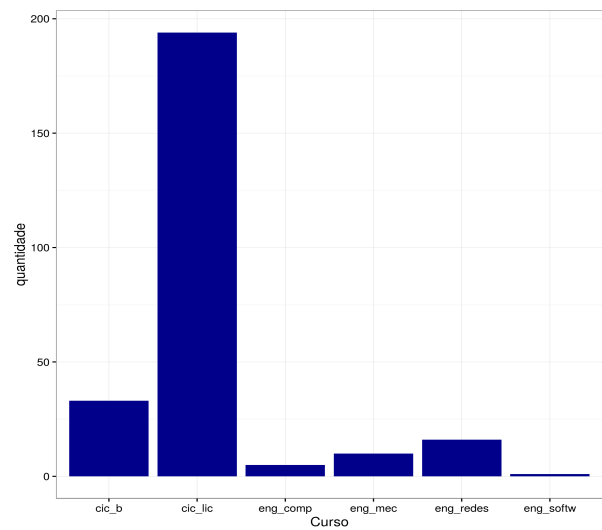
(a) Alunos Jovens da FT



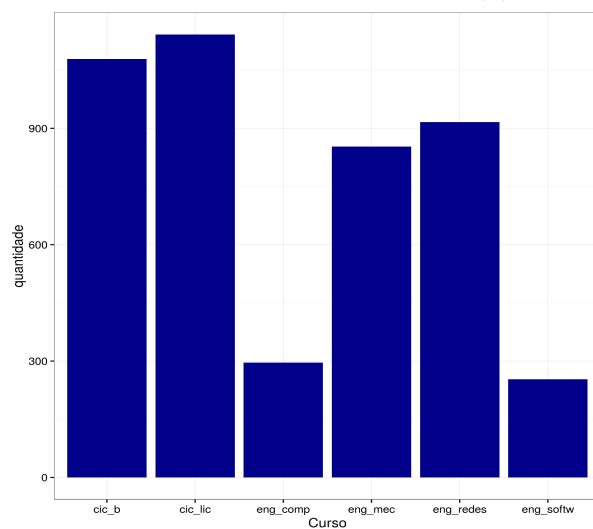
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação

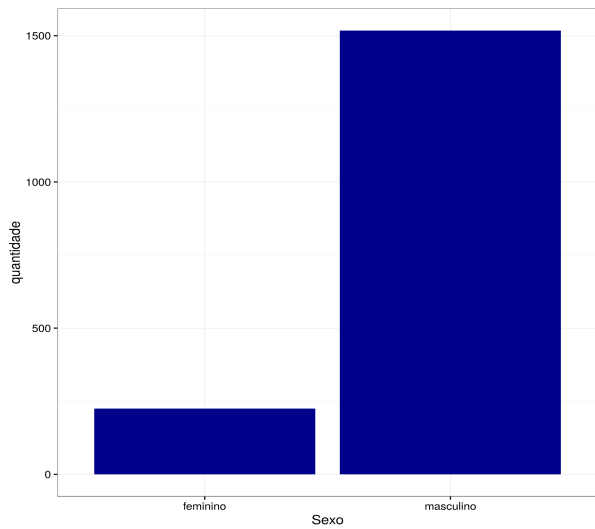


(d) Alunos Seniores

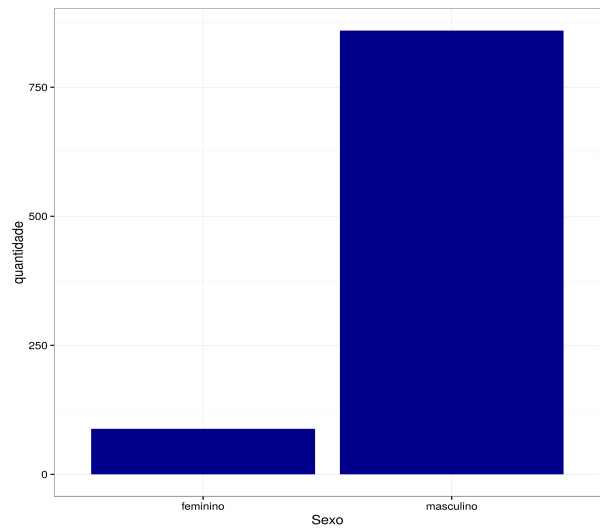


(e) Todos os Alunos

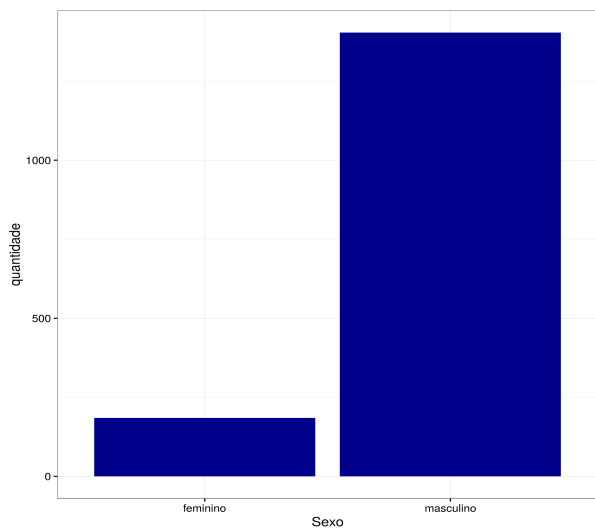
Figura 3.6: Atributo Curso, Conforme os Diferentes Modelos



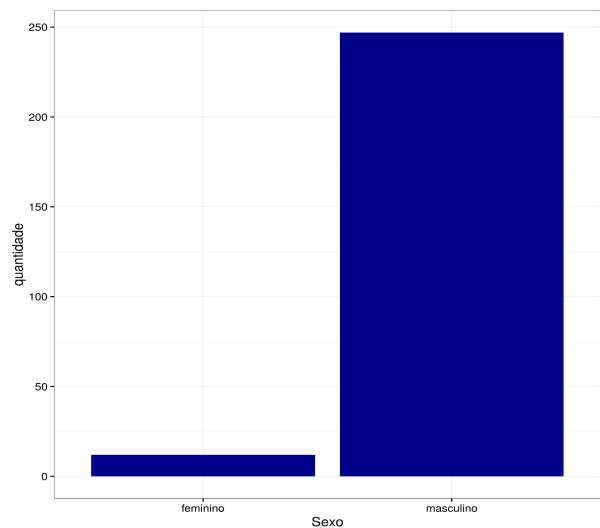
(a) Alunos Jovens da FT



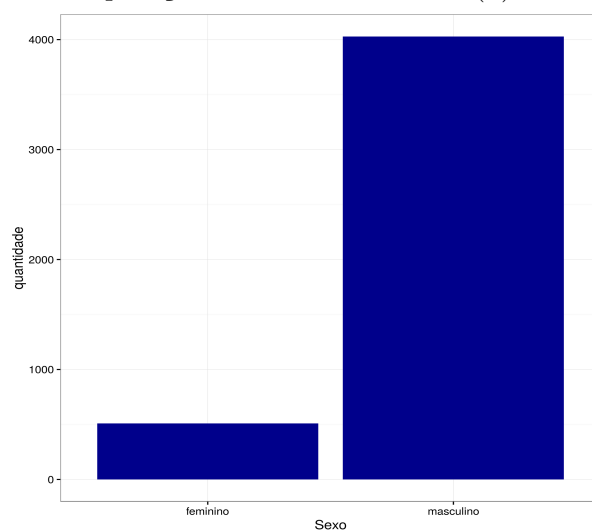
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação

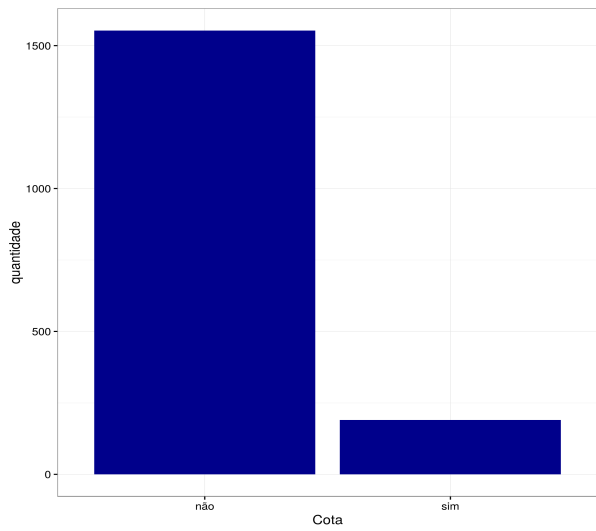


(d) Alunos Seniores

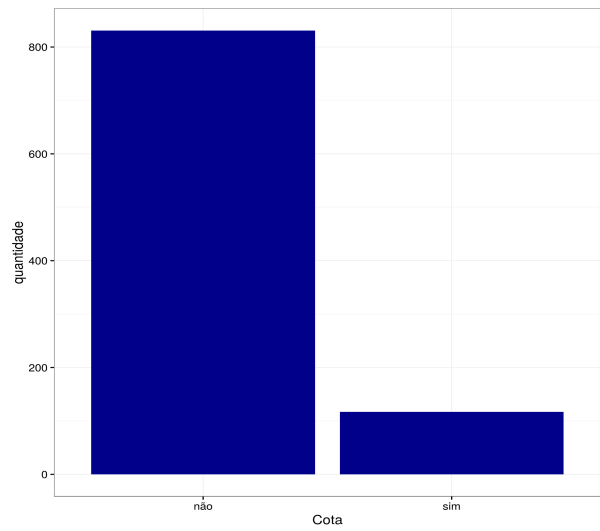


(e) Todos os Alunos

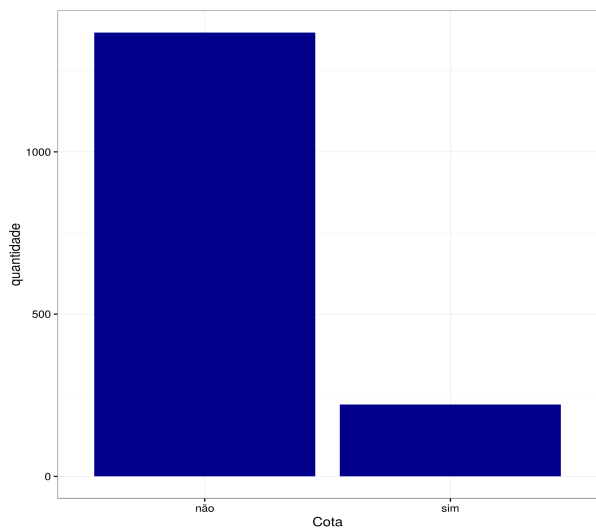
Figura 3.7: Atributo Sexo, Conforme os Diferentes Modelos



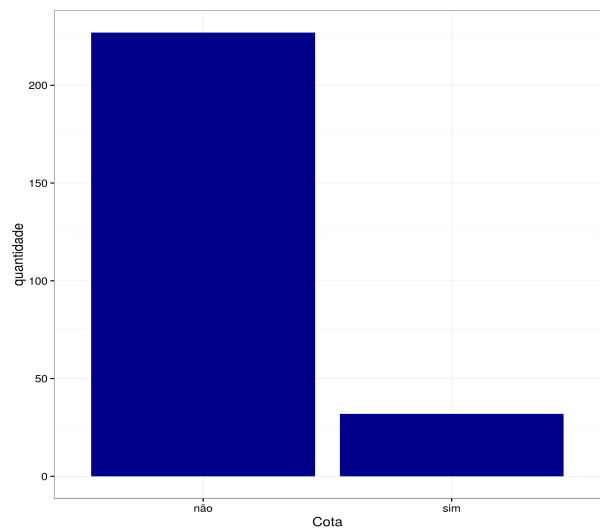
(a) Alunos Jovens da FT



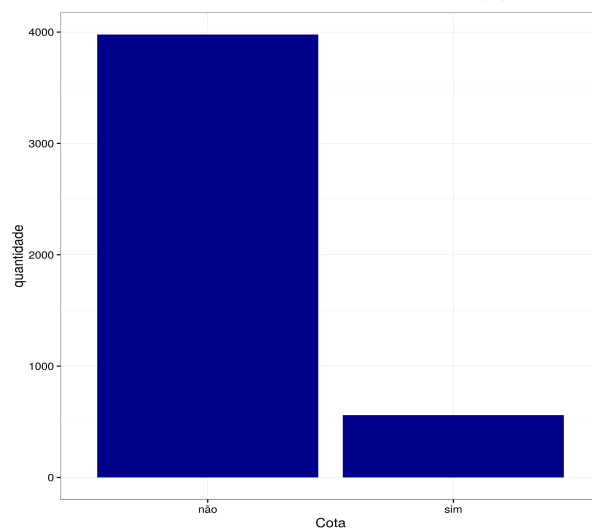
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação

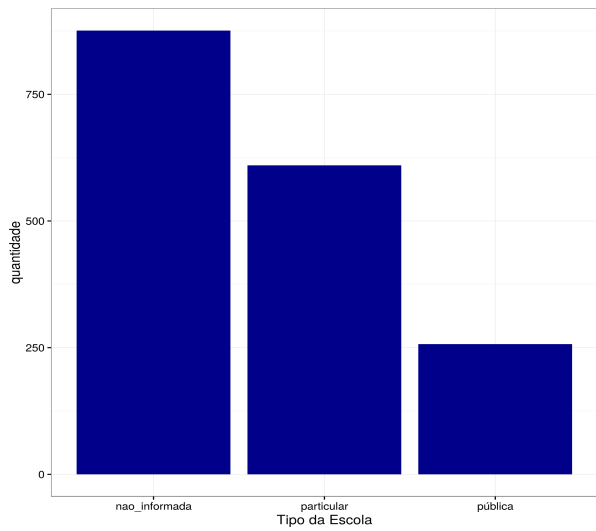


(d) Alunos Seniores

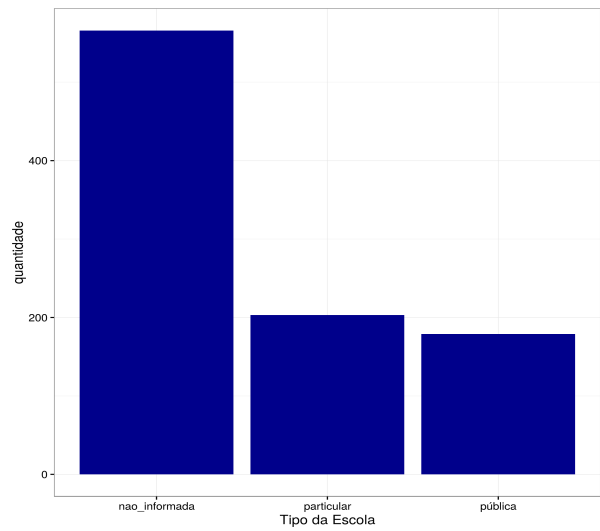


(e) Todos os Alunos

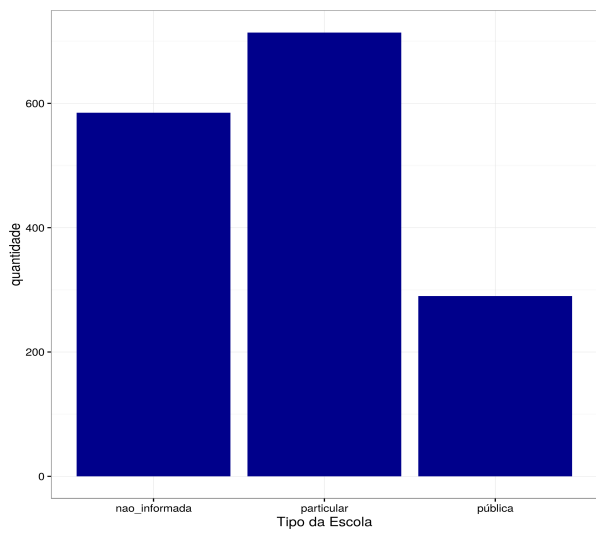
Figura 3.8: Atributo Cota, Conforme os Diferentes Modelos



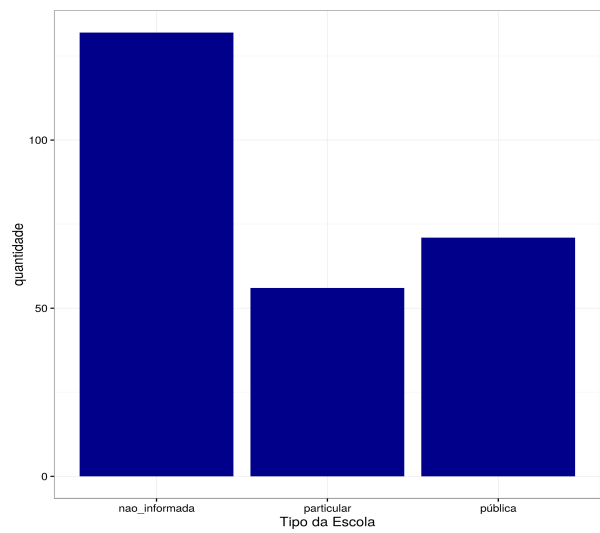
(a) Alunos Jovens da FT



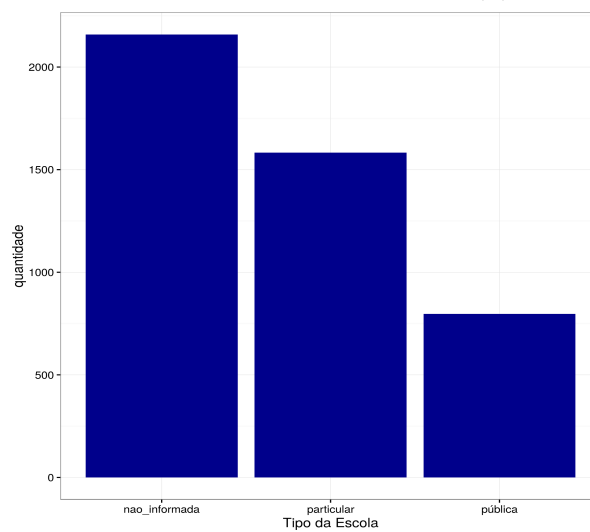
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação



(d) Alunos Seniores



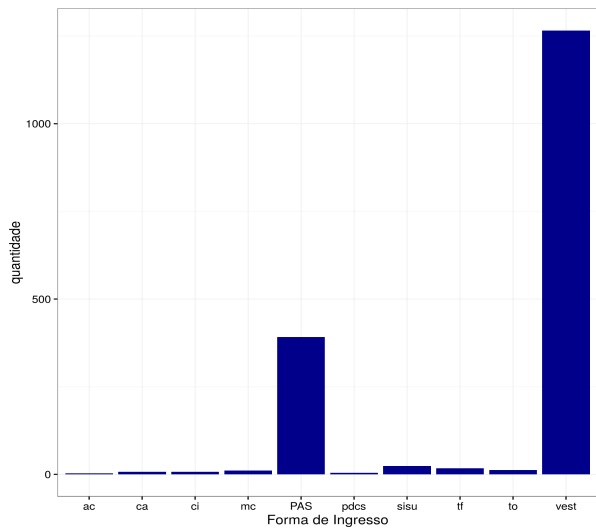
(e) Todos os Alunos

Figura 3.9: Atributo Tipo da Escola, Conforme os Diferentes Modelos

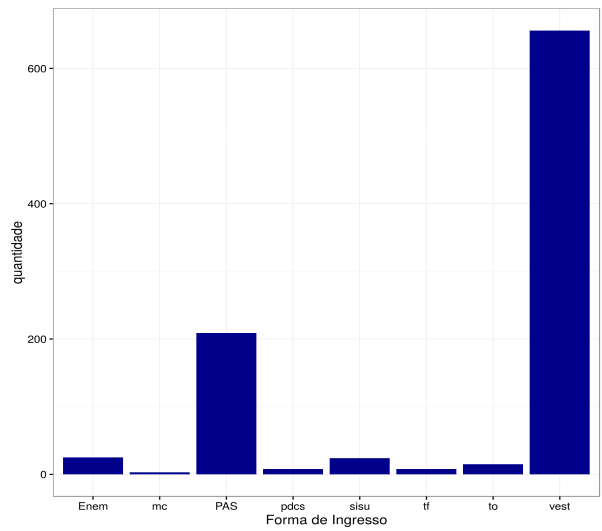
Por questões de legibilidade, as legendas nos gráficos foram encurtadas. Seu significado é apresentado a seguir:

- **vest:** Ingresso via Vestibular
- **ci:** Ingresso via Convênio-Int
- **to:** Ingresso via Transferência Obrigatória
- **ac:** Ingresso via Acordo Cultural PEC
- **ca:** Ingresso via Convênio Andifes
- **mc:** Ingresso via Matrícula Cortesia
- **tf:** Ingresso via Transferência Facultativa
- **ppp:** Ingresso via PEC-G Peppfol
- **pdcs:** Ingresso pois é portador de diploma de curso superior
- **vmc:** Ingresso via Vestibular para Mesmo Curso

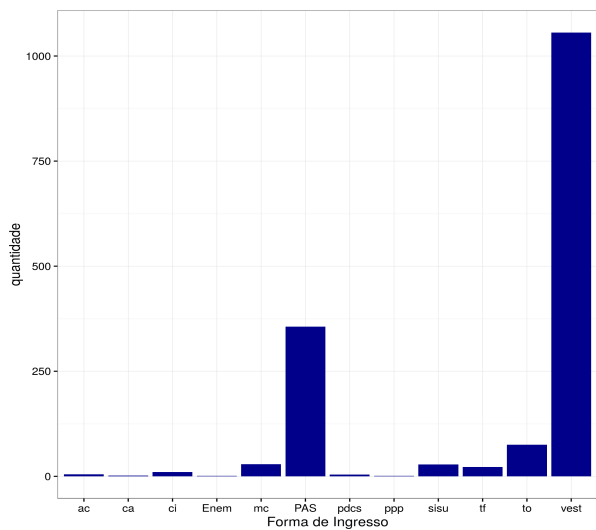




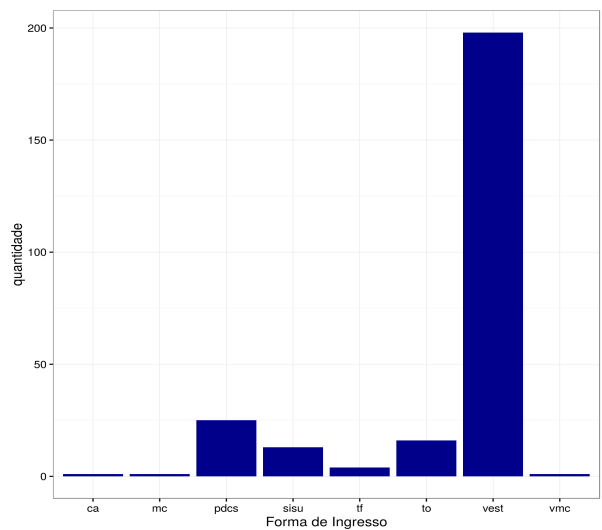
(a) Alunos Jovens da FT



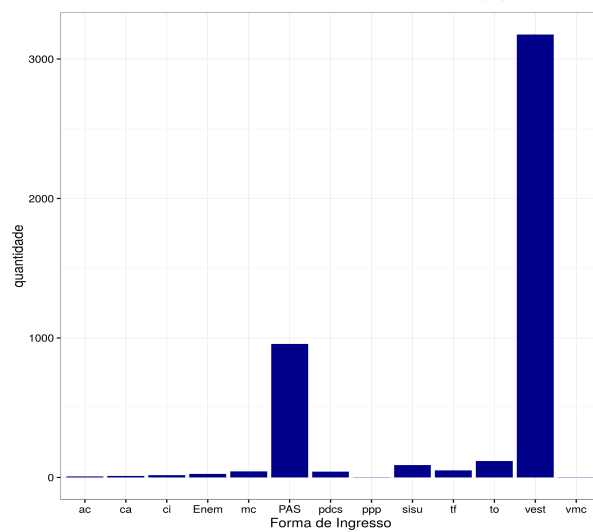
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação

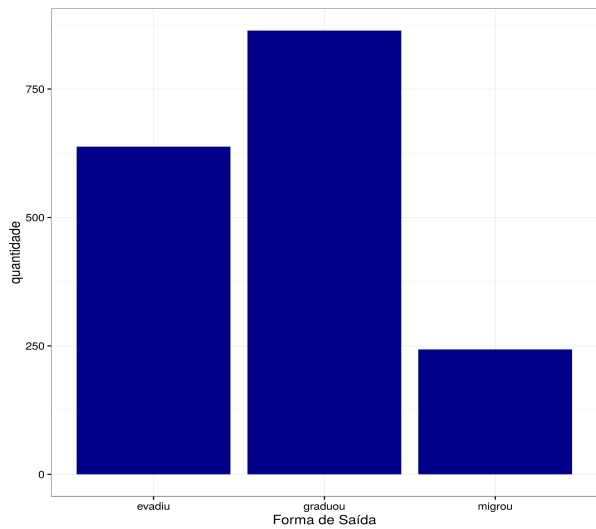


(d) Alunos Seniores

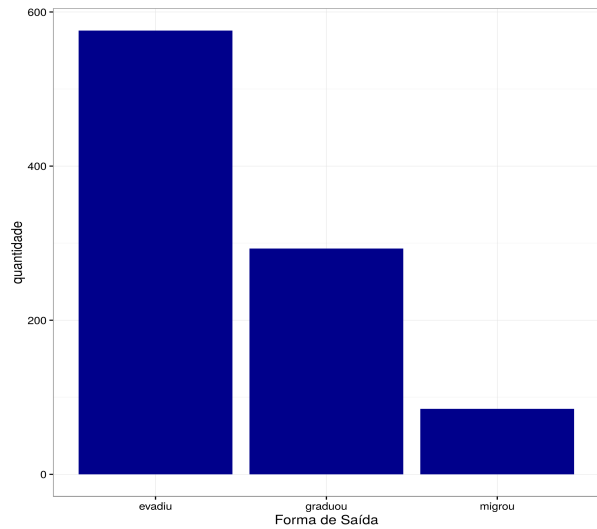


(e) Todos os Alunos

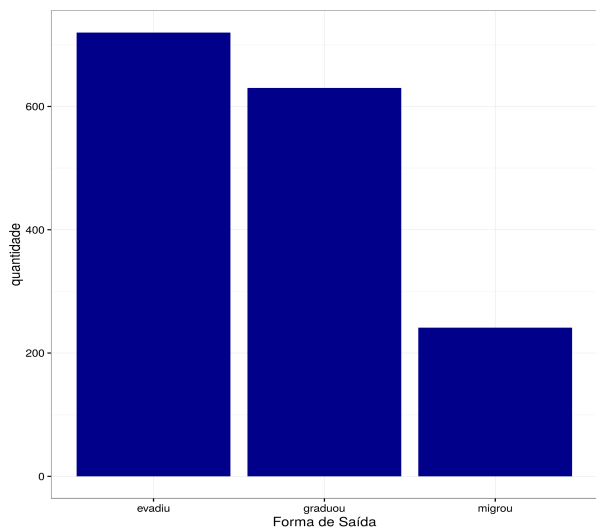
Figura 3.10: Atributo Forma de Ingresso, Conforme os Diferentes Modelos



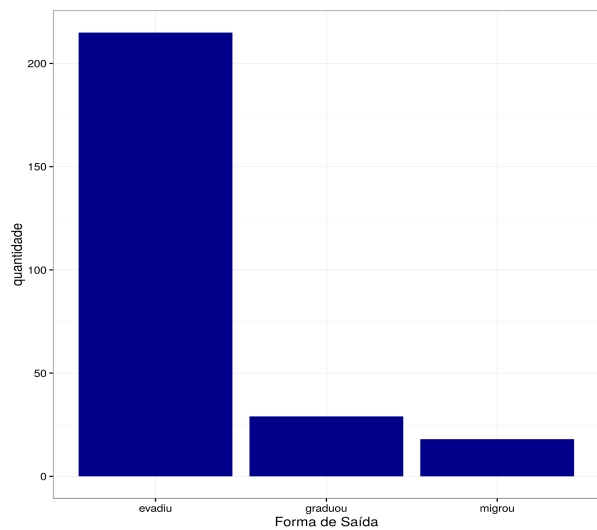
(a) Alunos Jovens da FT



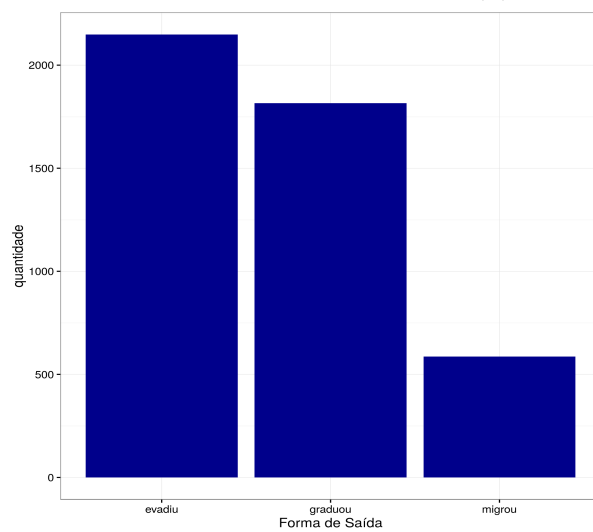
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação

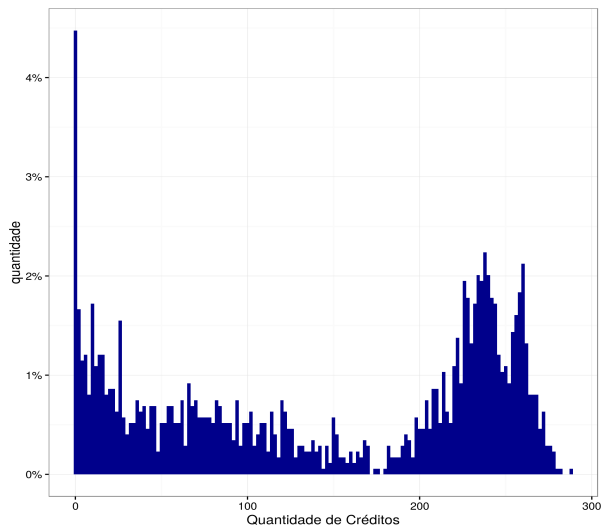


(d) Alunos Seniores

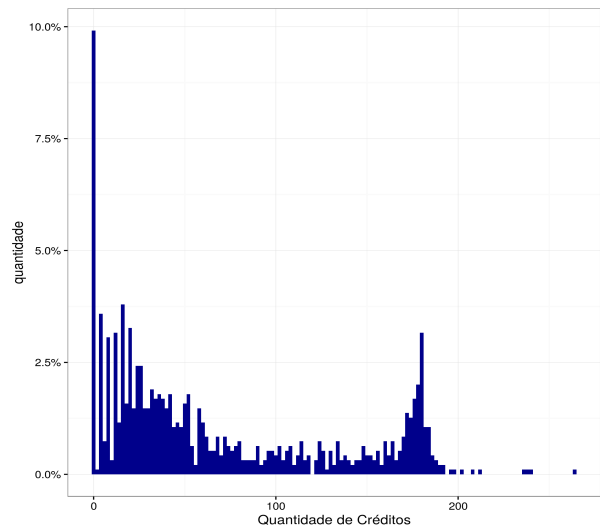


(e) Todos os Alunos

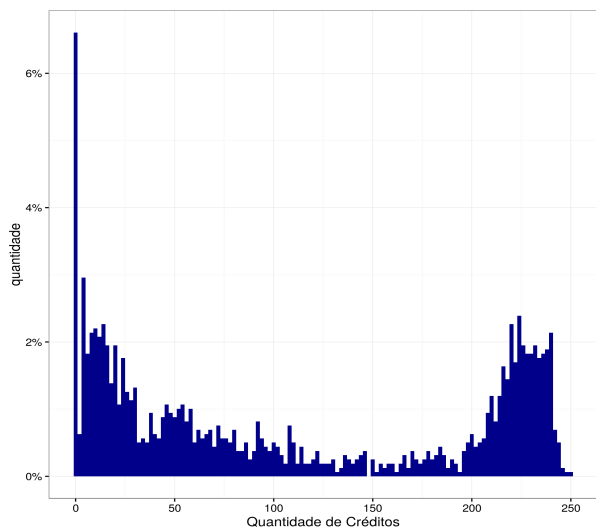
Figura 3.11: Atributo Forma de Saída, Conforme os Diferentes Modelos



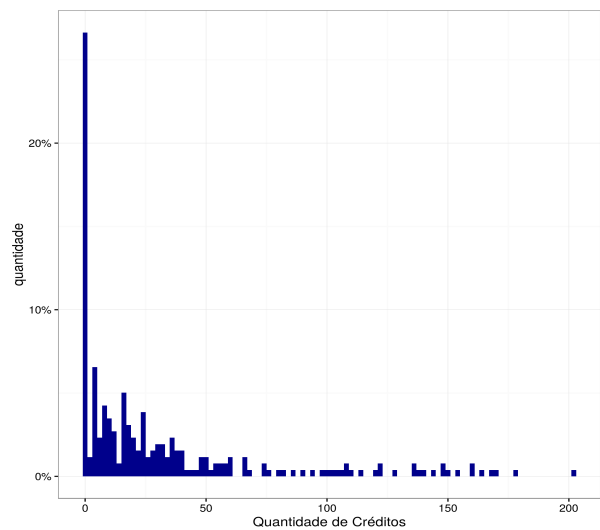
(a) Alunos Jovens da FT



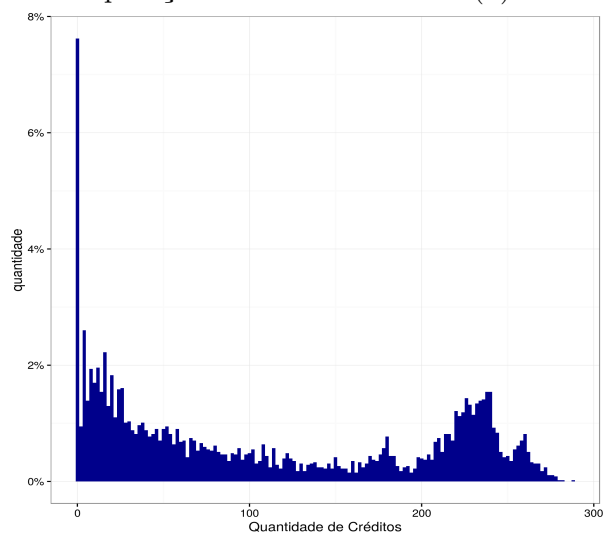
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação

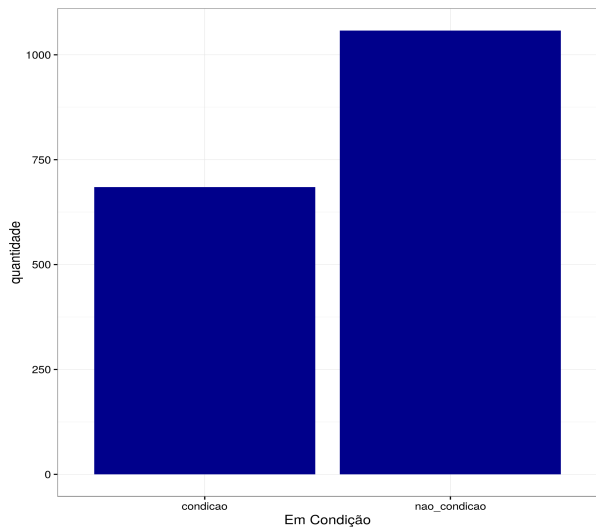


(d) Alunos Seniores

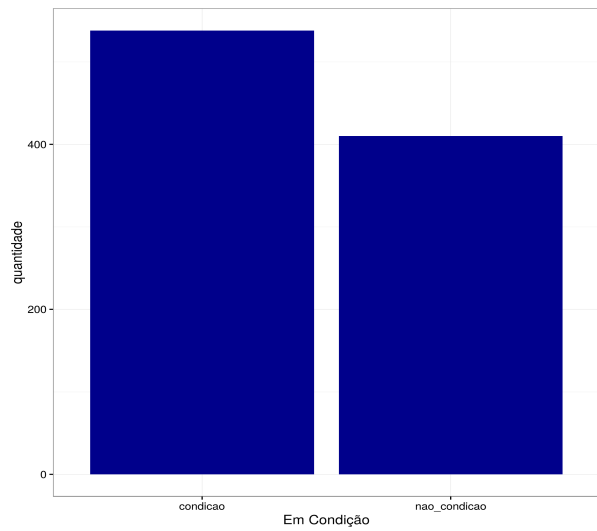


(e) Todos os Alunos

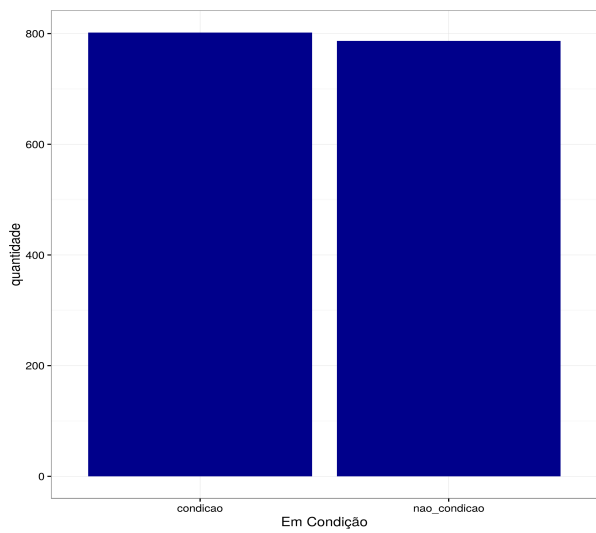
Figura 3.12: Atributo Quantidade de Créditos, Conforme os Diferentes Modelos



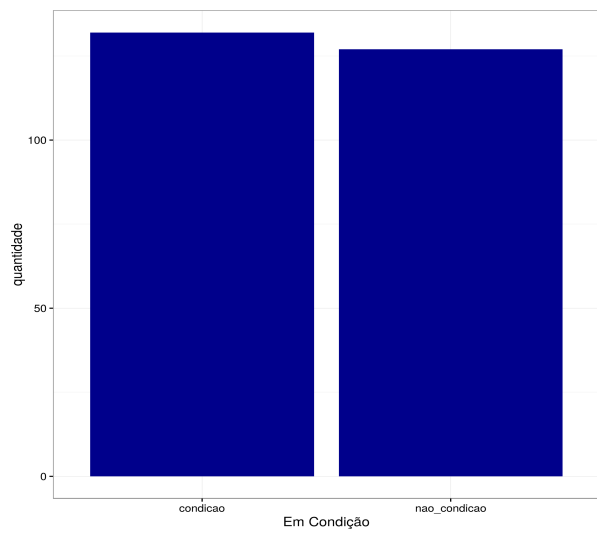
(a) Alunos Jovens da FT



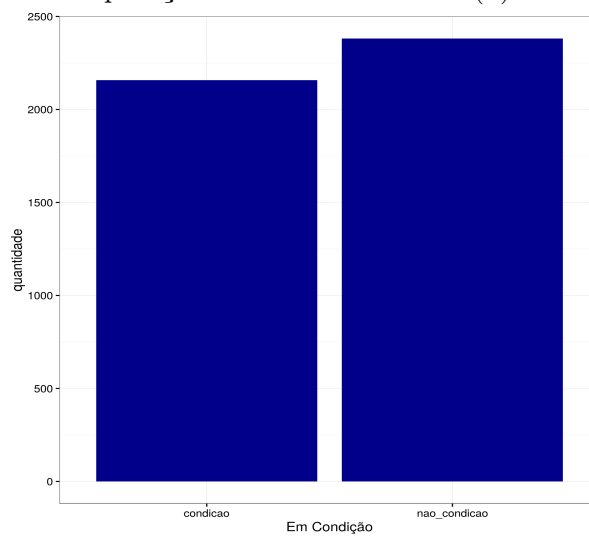
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação

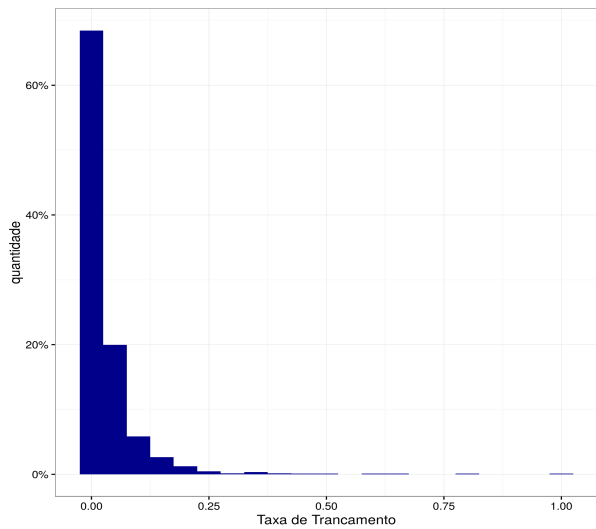


(d) Alunos Seniores

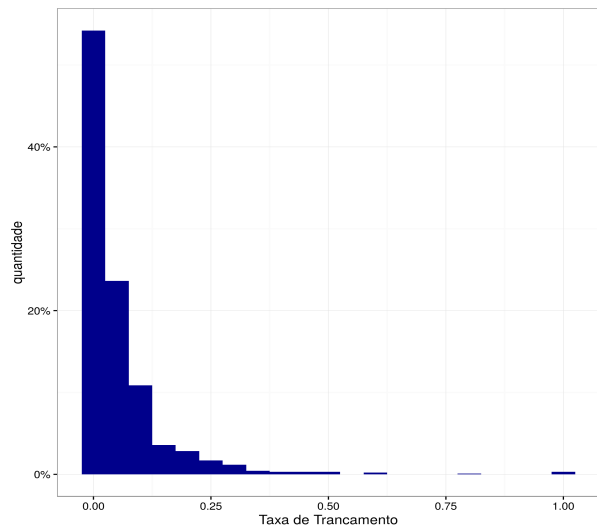


(e) Todos os Alunos

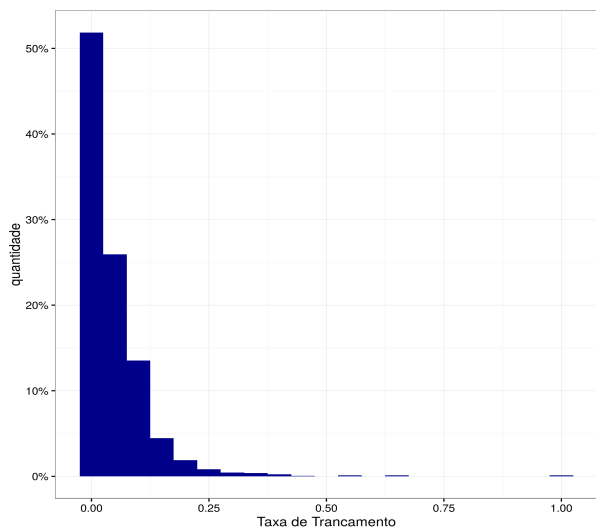
Figura 3.13: Atributo em condição, Conforme os Diferentes Modelos



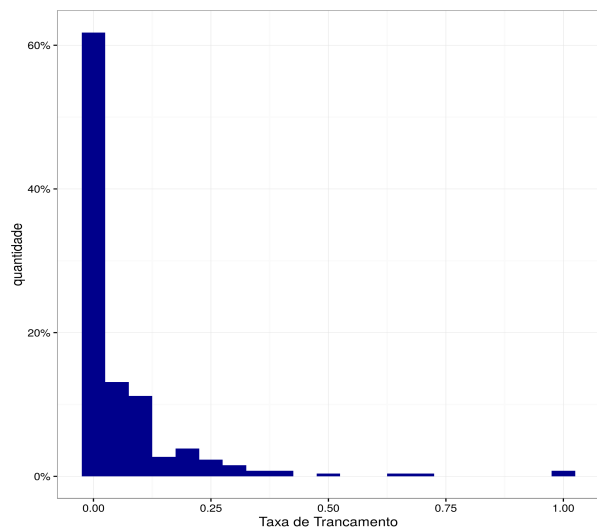
(a) Alunos Jovens da FT



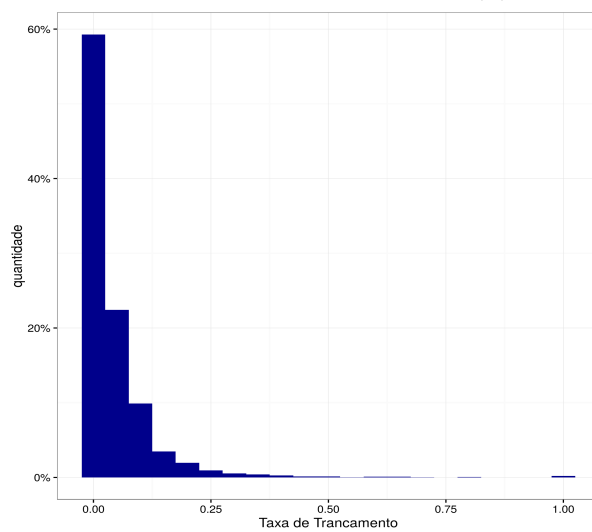
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação

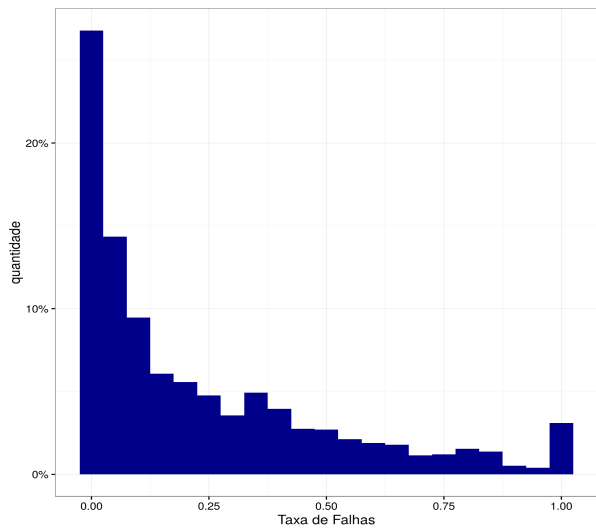


(d) Alunos Seniores

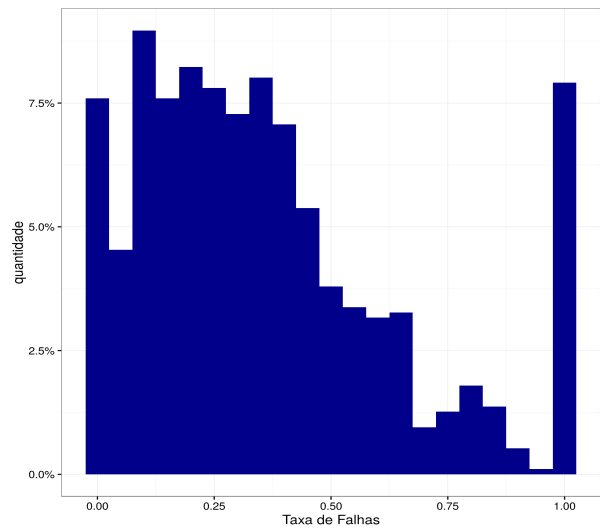


(e) Todos os Alunos

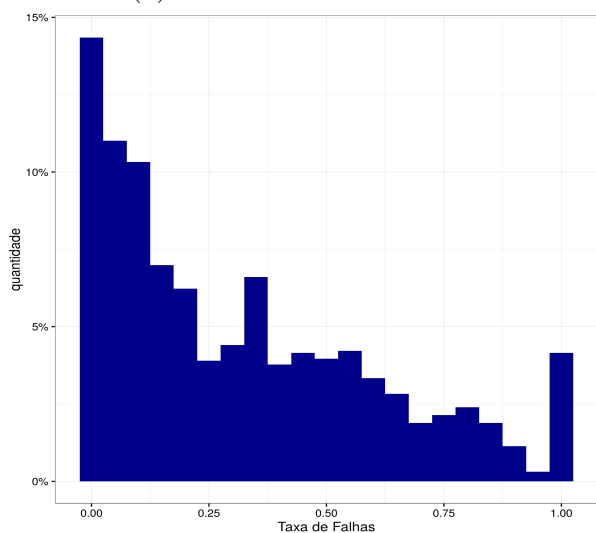
Figura 3.14: Atributo Taxa de Trancamento, Conforme os Diferentes Modelos



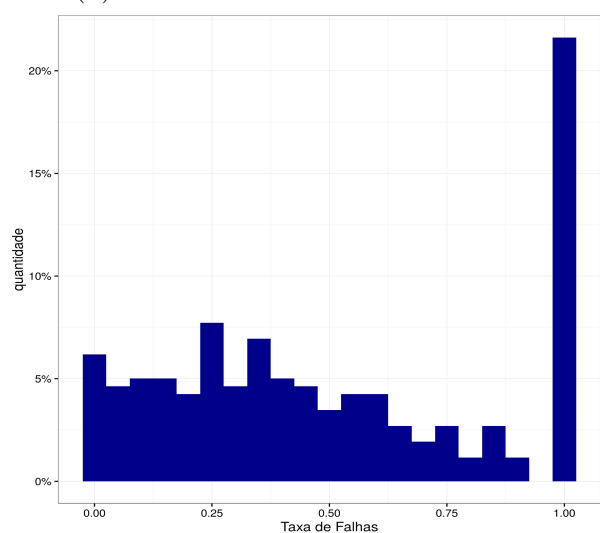
(a) Alunos Jovens da FT



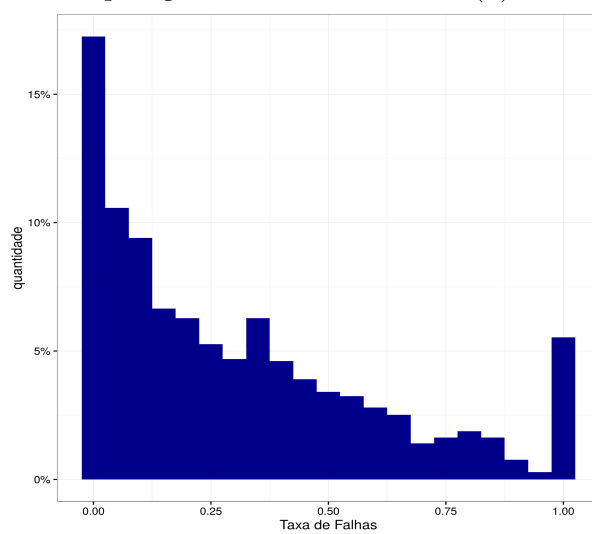
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação

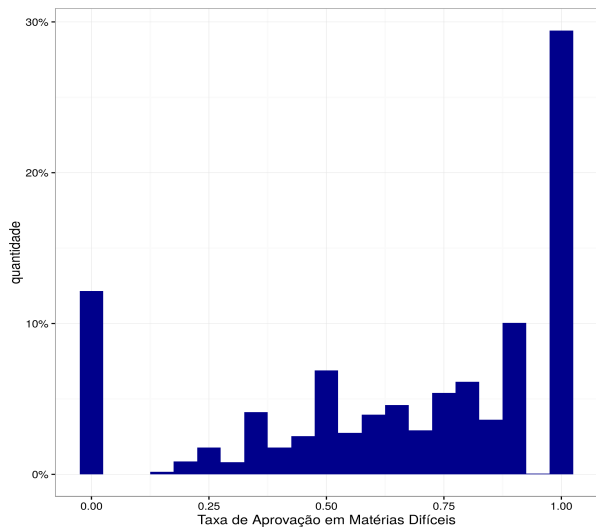


(d) Alunos Seniores

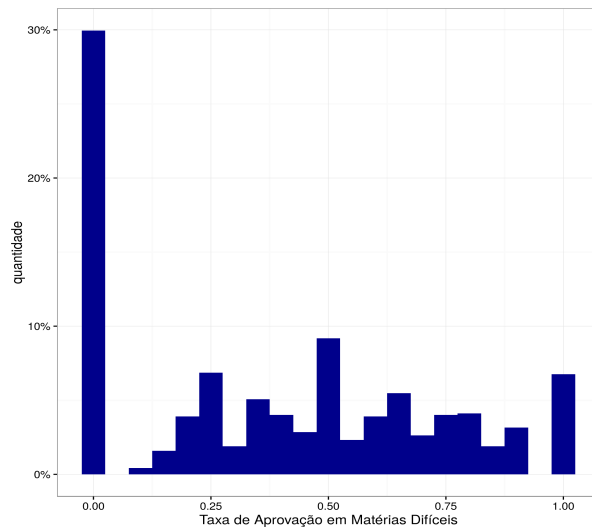


(e) Todos os Alunos

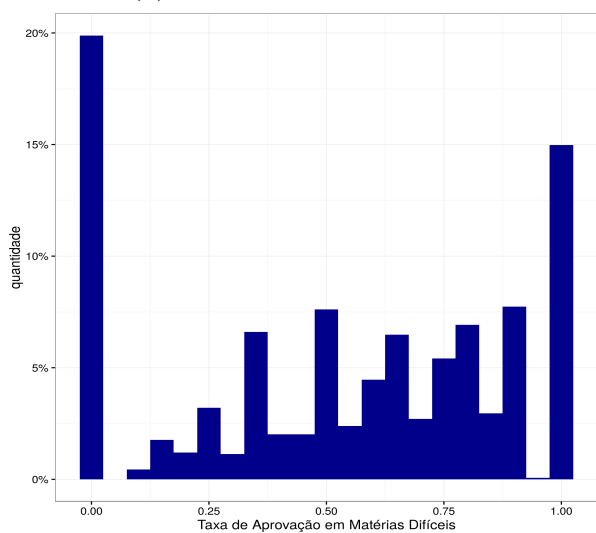
Figura 3.15: Atributo Taxa de Falhas, Conforme os Diferentes Modelos



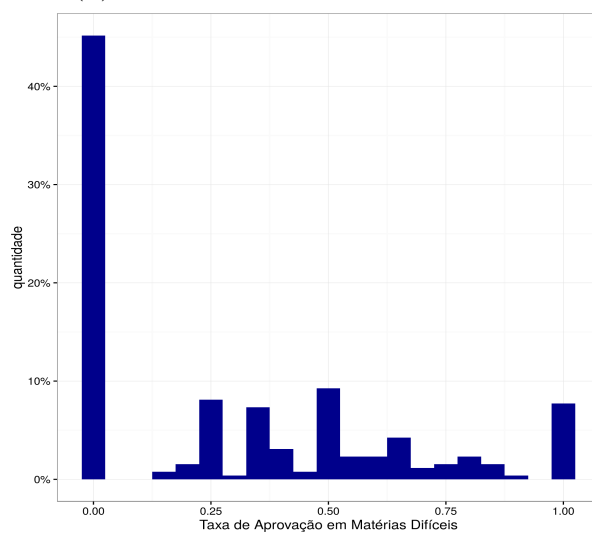
(a) Alunos Jovens da FT



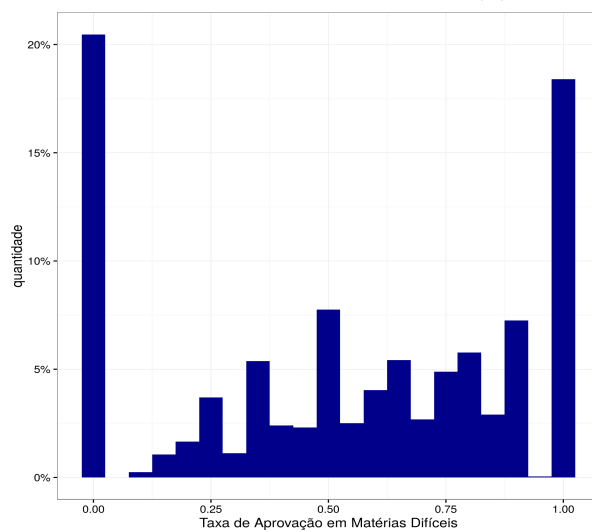
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação

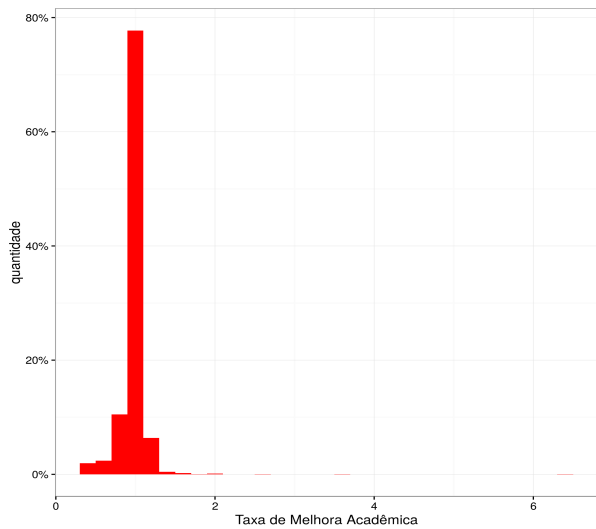


(d) Alunos Seniores

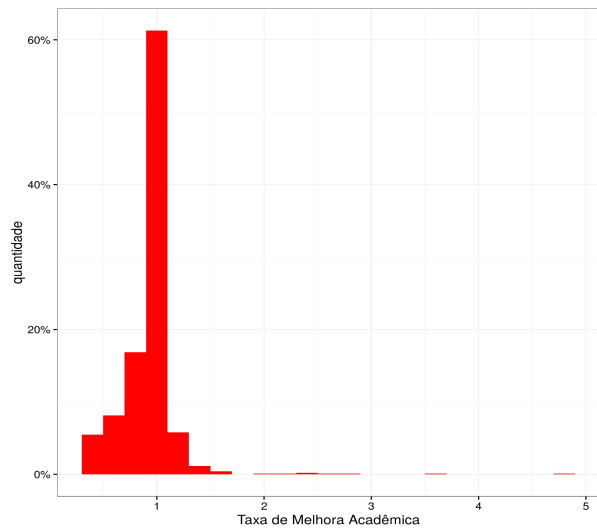


(e) Todos os Alunos

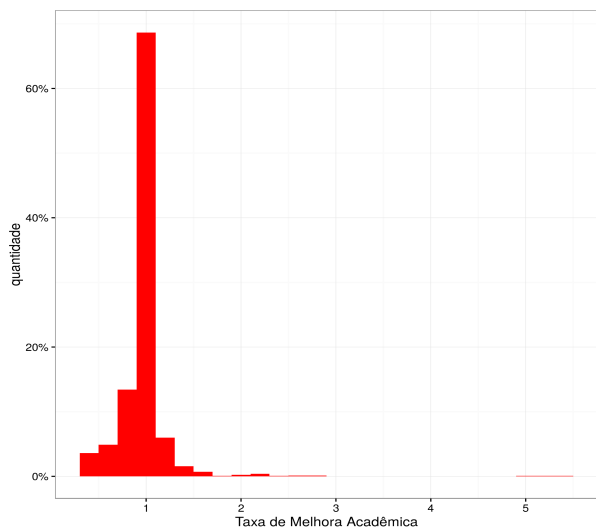
Figura 3.16: Atributo Taxa de Aprovação em Matérias Difíceis, Conforme os Diferentes Modelos



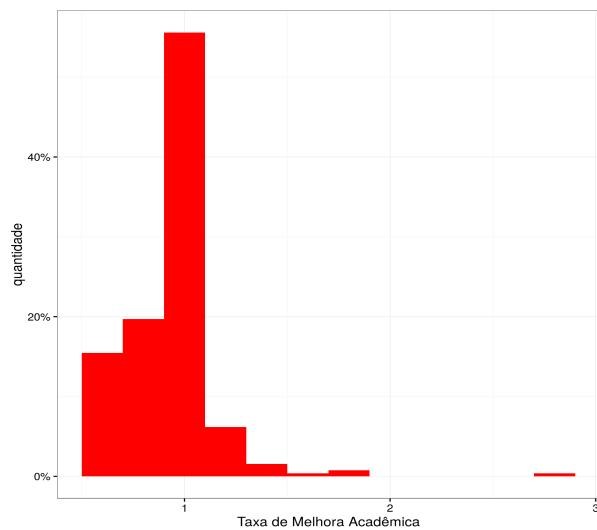
(a) Alunos Jovens da FT



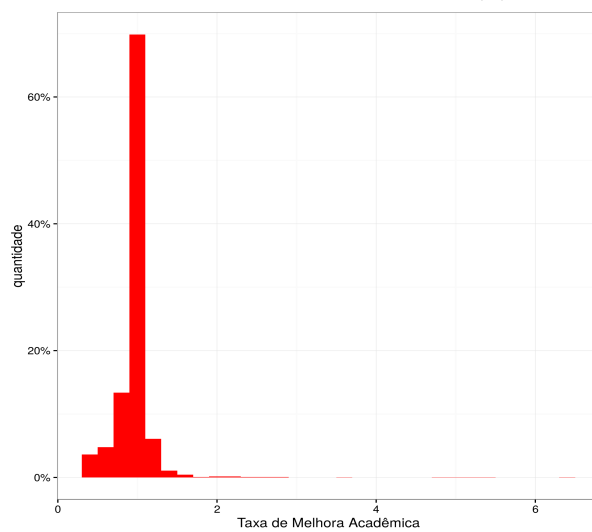
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação



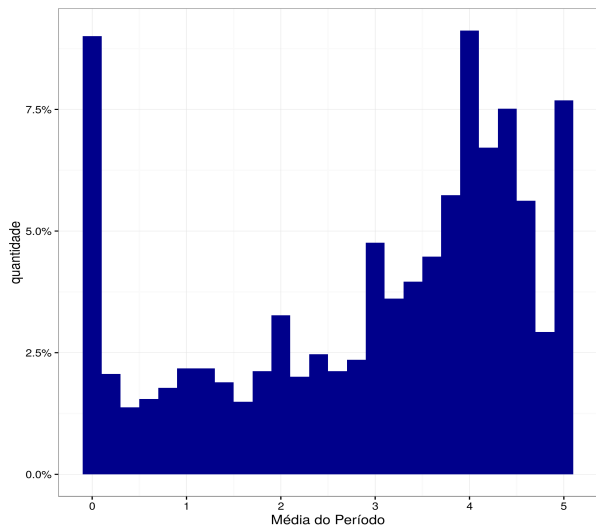
(d) Alunos Seniores



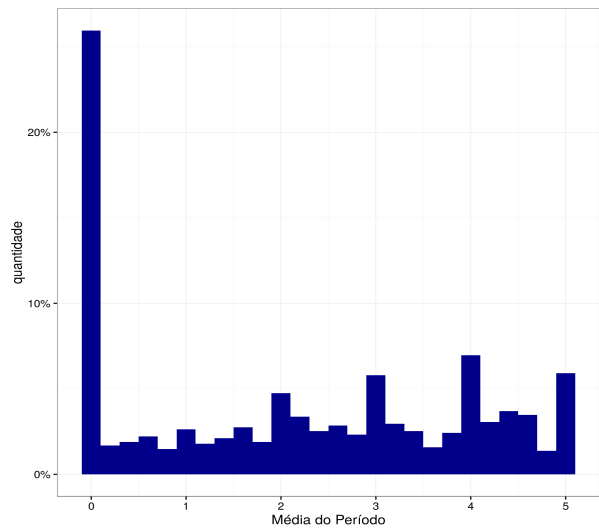
(e) Todos os Alunos

Figura 3.17: Atributo Taxa de Melhora, Conforme os Diferentes Modelos

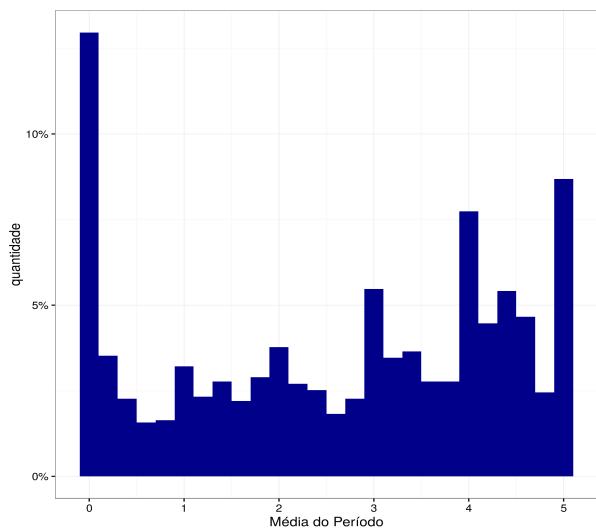




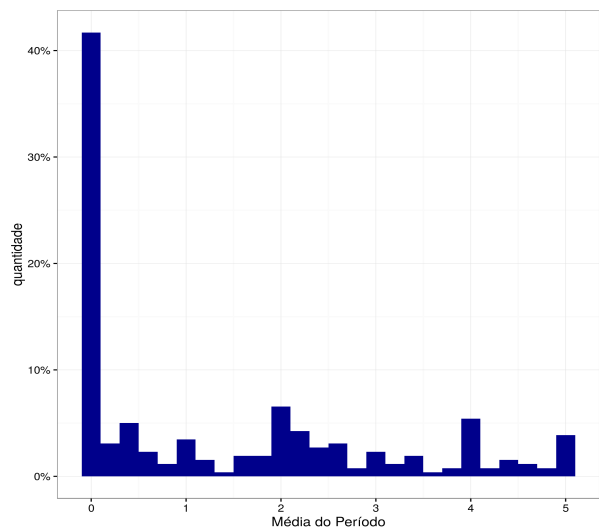
(a) Alunos Jovens da FT



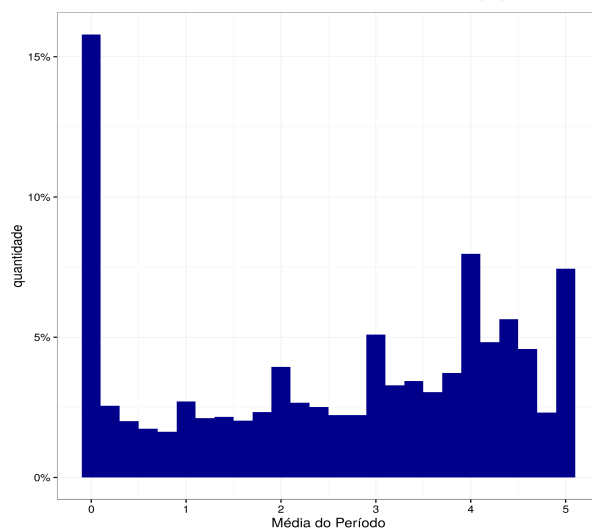
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação

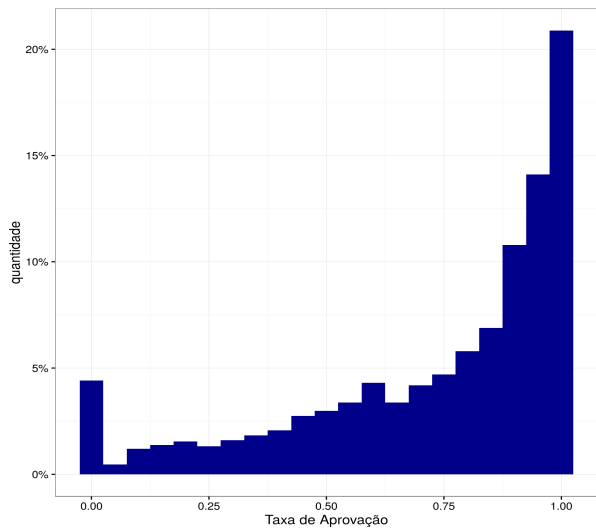


(d) Alunos Seniores

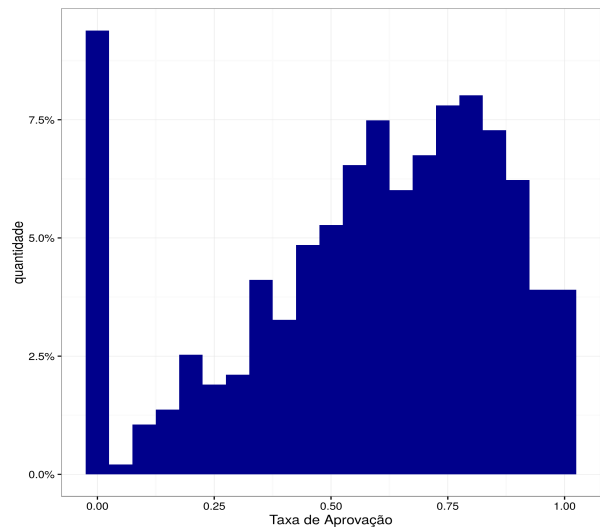


(e) Todos os Alunos

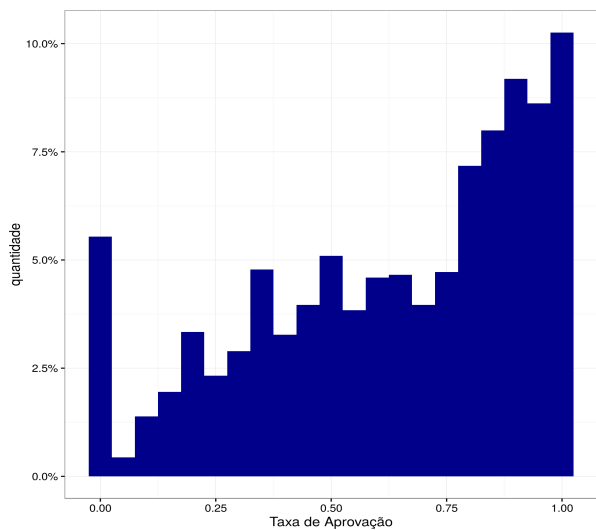
Figura 3.18: Atributo Média do Período, Conforme os Diferentes Modelos



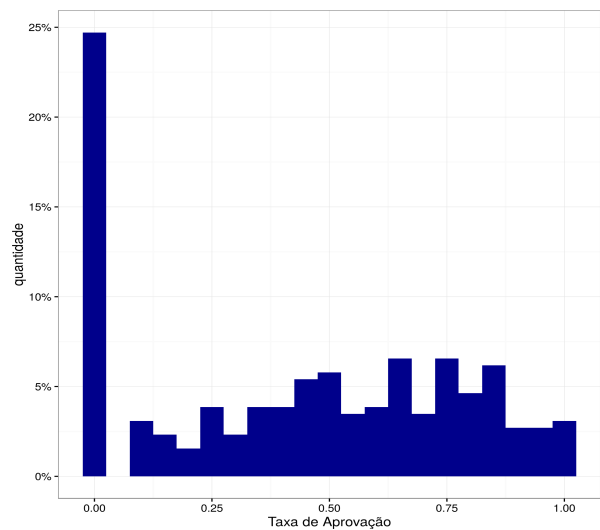
(a) Alunos Jovens da FT



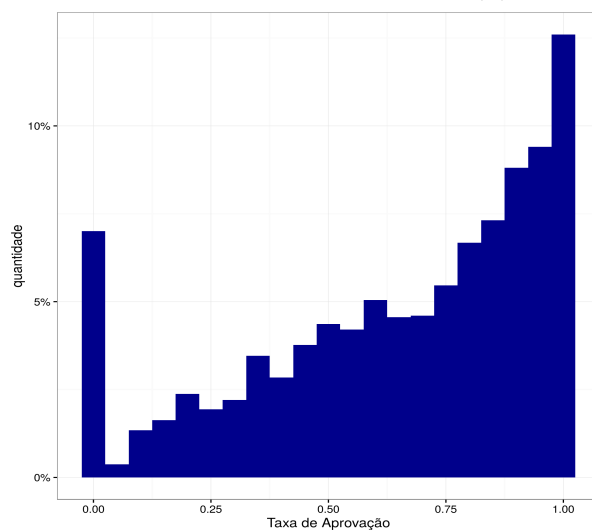
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação

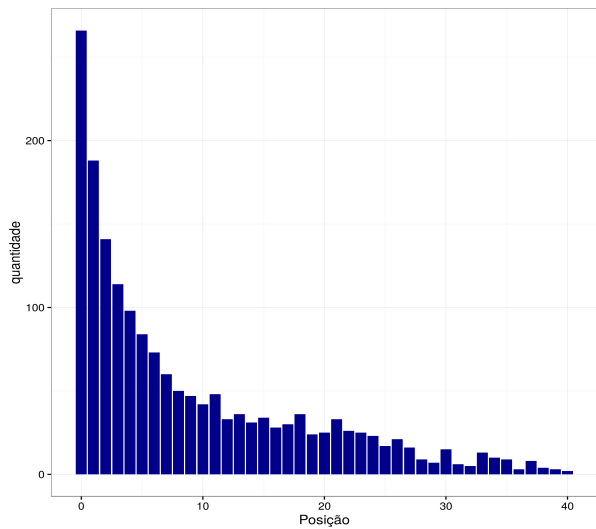


(d) Alunos Seniores

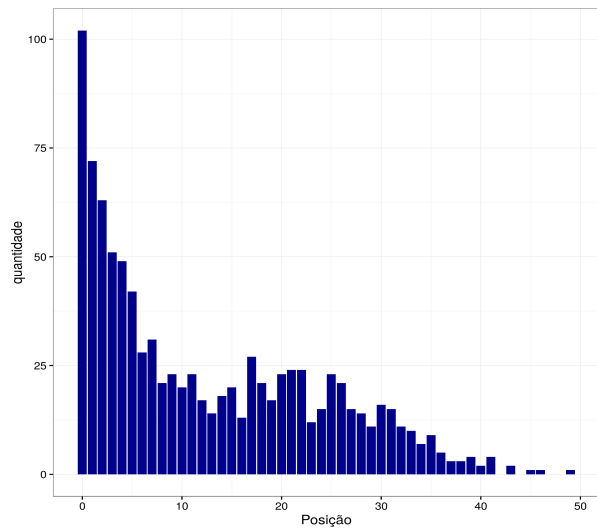


(e) Todos os Alunos

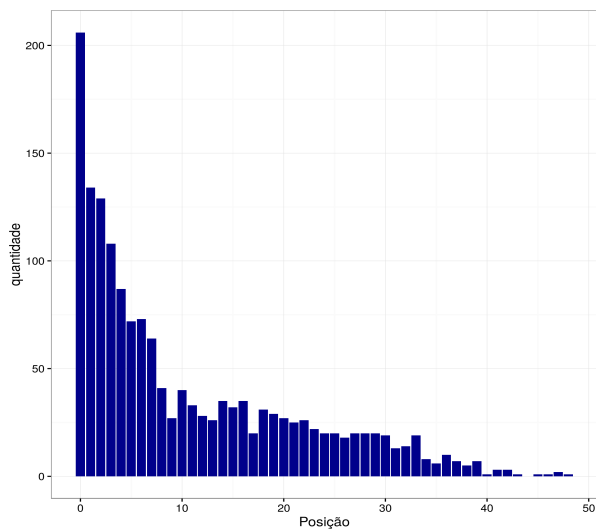
Figura 3.19: Atributo Taxa de Aprovação, Conforme os Diferentes Modelos



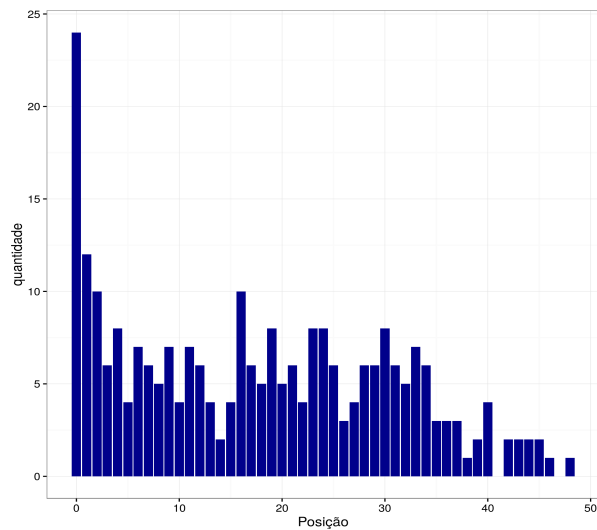
(a) Alunos Jovens da FT



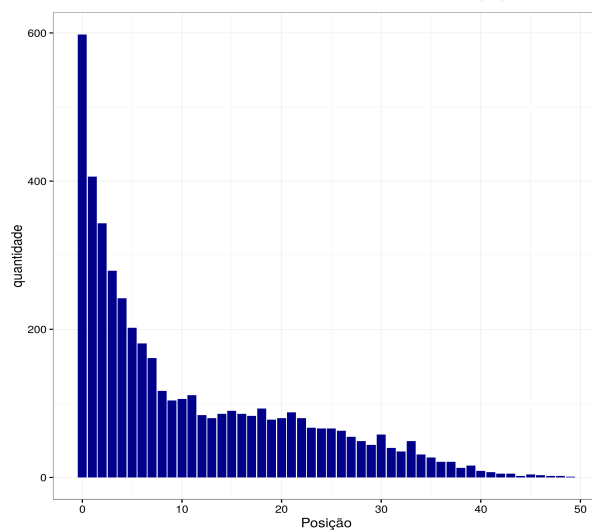
(b) Alunos Jovens da Licenciatura



(c) Alunos Jovens da Computação



(d) Alunos Seniores



(e) Todos os Alunos

Figura 3.20: Atributo Posição, Conforme os Diferentes Modelos

### 3.7 Eliminação de Atributos Relacionados ou Irrelevantes

Na área de aprendizagem de máquina, uma etapa fundamental é a da seleção de atributos [18], onde deve-se realizar a eliminação de atributos que estejam muito relacionados entre si e também a eliminação de atributos irrelevantes.

Avaliou-se o grau de relacionamento entre os diversos atributos por meio do coeficiente de correlação de Kendall. Optou-se por eliminar uma variável de cada par que apresentasse mais de 80% de correlação. Os atributos taxa de aprovação e taxa de reprovação apresentaram, para todas as bases de dados, alto relacionamento. Assim sendo, eliminou-se o atributo taxa de reprovação de análises posteriores.

Para descobrir os atributos irrelevantes, dentro daqueles originalmente pensados, foram usadas árvores de decisão: os atributos que não apareceram nas árvores de decisão foram eliminados das fases posteriores. Esse procedimento foi feito para cada uma das quatro bases de dados. Para os alunos jovens da licenciatura, obteve-se que o atributo curso não era relevante (o que era esperado já que todos os alunos desse grupo têm o mesmo curso). Para os alunos seniores, obteve-se que os atributos curso, cota e taxa de trancamento são irrelevantes. Para as outras bases de dados, nenhum atributo foi classificado como irrelevante.

### 3.8 Divisão em Treino e Teste

Como tradicionalmente ocorre no domínio de análise preditiva, houve a separação dos dados entre dados de treino e dados de teste. Os dados abrangiam alunos que entraram no período de 2000 até 2016 e já saíram da universidade. Aqueles alunos que entraram antes de 2010 formaram o conjunto dos dados de treino, enquanto que os alunos que entraram de 2010 em diante formaram o conjunto dos dados de teste. A separação dos dados de parte dos dados de treino para validação ocorreu somente na etapa do ajuste de parâmetros, sendo descrita na Seção 3.11.

### 3.9 A Divisão em Semestres

Para o problema de negócio considerado na pesquisa, é necessário que o sistema predictor seja capaz de calcular o risco de alunos evadirem tanto para estudantes nos semestres iniciais do curso quanto para estudantes mais adiantados. Relacionado a isso, tem-se que alguns atributos dos alunos, como por exemplo a taxa de aprovação, mudam a cada semestre. Tal fato deve ser considerado na hora de treinar os modelos.

Pensando nisso, os modelos de mineração de dados são induzidos separadamente para cada semestre. Assim, inicialmente os modelos são induzidos com os dados relativos ao 1º semestre de cada aluno do conjunto de treino e são testados com os dados relativos ao 1º semestre de cada aluno do conjunto de teste. Após isso, repete-se tal procedimento para os dados dos alunos relativos ao 2º semestre e assim sucessivamente.

Caso esteja-se estudando um semestre para o qual o aluno em questão já saiu da UnB (por exemplo, está se estudando a capacidade do sistema predictor para alunos no 10º semestre e o aluno em questão saiu da UnB ao fim do 8º semestre) tal aluno não entra no conjunto de treino/teste para o semestre considerado.

### 3.10 Algoritmos de Aprendizagem de Máquina Estudados e Retroalimentação

Os algoritmos de aprendizagem de máquina utilizadas para a predição de alunos em risco de evasão foram: *Naive Bayes*, *random forests*, rede neural, regressor linear e SVR. Utilizou-se a biblioteca `scikit-learn` (versão 0.18.1) [19], da linguagem de programação Python.

Para cada um desses algoritmos, estudou-se se utilizar retroalimentação poderia melhorar o desempenho. A retroalimentação funcionaria da seguinte forma: o modelo de aprendizagem de máquina, na hora de tentar prever o desempenho para um semestre  $x$ , receberia as chances do aluno graduar/evadir/migrar calculada por esse mesmo modelo para o semestre  $x - 1$ . Para cada modelo estudado, analisou-se seu desempenho com e sem retroalimentação.

### 3.11 Ajuste de Parâmetros

Estimou-se quais seriam os melhores parâmetros para os seguintes algoritmos de aprendizagem de máquina: *Naive Bayes*, rede neural e SVR. Para os demais, seguiu-se as configurações padrão da biblioteca `scikit-learn`. Para isso, utilizou-se validação, como descrito a seguir. Inicialmente, dividiu-se o conjunto que não era de teste (consistindo dos alunos que entraram antes de 2010) em dois subconjuntos: o conjunto de treino (alunos que entraram antes de 2007) e o conjunto de validação (alunos que entraram de 2007 em diante). Depois, cada configuração de parâmetros de cada método treinava no conjunto de treino e tinha seu desempenho avaliado no conjunto de validação.

Por fim, para escolher a melhor configuração de cada modelo de aprendizagem de máquina, considerou-se, para cada configuração, o desempenho obtido semestre a semestre.

Assumindo uma distribuição normal, obtinha-se o intervalo de confiança do desempenho do modelo sob à configuração sendo testada. Para cada modelo, escolhia-se a configuração com menor intervalo de confiança, desde que tal intervalo tivesse intersecção não vazia com o intervalo de confiança da configuração com melhor desempenho. Adotou-se um intervalo de confiança de 95%. As configurações obtidas para cada modelo são mostradas na Seção 4.1.

### 3.11.1 Ajuste de Parâmetros Para Redes Neurais

Estimou-se primeiramente a quantidade de neurônios para a camada oculta, testando o desempenho das ANNs com 12, 24, 36 e 100 neurônios. Em seguida, testou-se a taxa de aprendizagem para a rede neural, experimentando os valores 0.001 (padrão da biblioteca), 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 e 1.0. Os resultados são mostrados na Seção 4.1.1.

### 3.11.2 Ajuste de Parâmetros Para SVR

Estimou-se primeiramente a melhor configuração para o tipo de *kernel* da SVR. Os tipos de *kernel* analisados foram linear, polinomial e RBF. Em seguida, variou-se o parâmetro de penalização  $C$  da SVR, analisando-se o desempenho da SVR quando  $C$  era 0,5 1 e 2. Os resultados são mostrados na Seção 4.1.2.

### 3.11.3 Estimativa de Parâmetros Para Naive Bayes

O único parâmetro estimado para o algoritmo de aprendizagem de máquina *Naive Bayes* foi o tipo de distribuição que os atributos seguiam. Testou-se o desempenho de tal modelo de aprendizagem de máquina assumindo-se que os atributos tinham as seguintes distribuições: gaussiana, multinomial e Bernoulli. Os resultados são mostrados na Seção 4.1.3.

## 3.12 Avaliação de Desempenho

Após selecionar as melhores configurações para cada algoritmo de aprendizagem de máquina, avaliou-se o desempenho de cada um dos modelos induzidos. Para isso, cada modelo foi induzido no conjunto de treino (composto por alunos que ingressaram antes de 2010) e teve seu desempenho avaliado no conjunto de teste (composto por alunos que ingressaram de 2010 em diante). Esse processo foi feito para cada um dos semestres estudados e funciona como explicado a seguir.

Cada modelo induzido pelo conjunto de treino de cada base de dados gera, para cada aluno em cada semestre ativo, uma tripla que indica a possibilidade do aluno concluir, evadir ou migrar. O maior valor da tripla é usado como sendo a previsão do modelo. Tal previsão é então comparada com o que realmente aconteceu ao aluno. Se a previsão condiz com o real, o modelo acerta; caso contrário, o modelo erra.

A métrica utilizada para avaliar o desempenho dos modelos foi a *F-measure*. Um determinado modelo tem, para uma determinada base de dados, vários valores de *F-measure* calculados, um para cada semestre. Feito isso, para sumarizar o desempenho do algoritmo para os alunos de teste de uma base de dados, toma-se a média das *F-measures* de cada semestre. Compararam-se os resultados dos modelos entre si, e também com o modelo ZeroR. Os resultados encontram-se na Seção 4.2.

A Figura 3.21 sumariza o processo descrito acima. Para cada semestre os dados de treino são utilizados para induzir modelos, que são então avaliados nos dados de teste. Esse processo de avaliação irá gerar, para cada semestre, um valor de *F-measure*. Por fim, de modo a sumarizar o desempenho do algoritmo, calcula-se a média das *F-measures*.

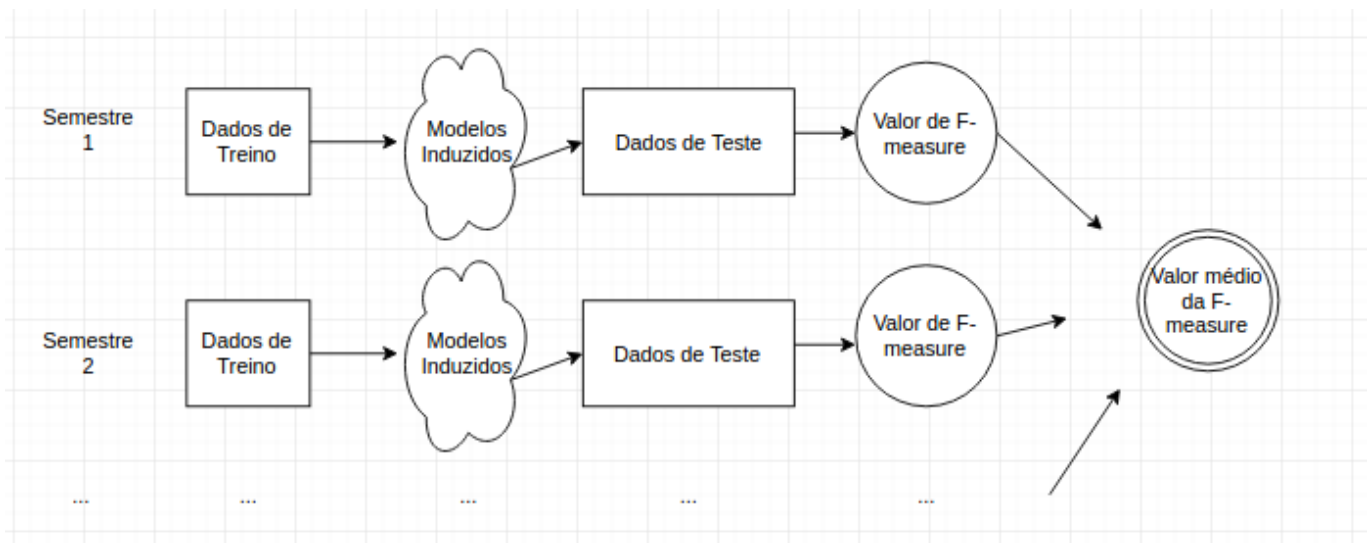


Figura 3.21: Diagrama Explicativo Mostrando Processo de Avaliação do Desempenho

# Capítulo 4

## Análise dos Resultados

Nesta capítulo, mostram-se as melhores configurações para os modelos de aprendizagem de máquina estudados. Em seguida, apresentam-se os resultados obtidos para as diversas técnicas de aprendizagem de máquina consideradas. Por fim, analisam-se os resultados obtidos.

### 4.1 Configurações Obtidas Para Modelos de Aprendizagem de Máquina

Nesta seção, analisa-se quais foram as melhores configurações obtidas para os seguintes métodos de mineração de dados: ANN, SVR e *Naive Bayes*.

#### 4.1.1 Configurações da ANN

Inicialmente, estudou-se qual seria a quantidade ótima de neurônios da camada escondida da rede neural. Depois, estudou-se qual seria o melhor valor para o parâmetro taxa de aprendizagem. Mostrou-se também o desempenho do melhor modelo no conjunto de validação. Os resultados estão disponíveis na Tabela 4.1.

Tabela 4.1: Melhor Escolha de Parâmetros Para ANN

Dados	Neurônios	Aprendizagem	F-measure
Alunos Jovens da FT	100	0.001	0.81
Alunos Jovens da Licenciatura	36	1.0	0.79
Alunos Jovens da Computação	36	0.001	0.77
Alunos Seniores	24	0.7	0.77



Tabela 4.2: Melhor Escolha de Parâmetros Para SVR

Dados	Kernel	Penalização	F-measure
Alunos Jovens da FT	linear	1.0	0.79
Alunos Jovens da Licenciatura	linear	1.0	0.81
Alunos Jovens da Computação	rbf	1.0	0.83
Alunos Seniores	linear	1.0	0.67

Tabela 4.3: Melhor Escolha de Parâmetros Para Naive Bayes

Dados	Atributos	F-measure
Alunos Jovens da FT	Gaussiana	0.63
Alunos Jovens da Licenciatura	Bernoulli	0.78
Alunos Jovens da Computação	Multinomial	0.72
Alunos Seniores	Gaussiana	0.64

Os resultados da Tabela 4.1 evidenciam que a melhor configuração, tanto para o número de neurônios quanto para a taxa de aprendizagem, varia conforme a base de dados em questão.

#### 4.1.2 Configurações da SVR

Estudou-se como o tipo de *kernel* e o parâmetro de penalização poderiam ser escolhidos de modo a obter uma boa configuração para a SVR. Mostrou-se também o desempenho do melhor modelo no conjunto de validação. Os resultados são apresentados na Tabela 4.2.

Os resultados da Tabela 4.2 mostram que as melhores configurações não dependem muito da base de dados com a qual estamos trabalhando. Em três das quatro bases de dados, o tipo de *kernel* escolhido foi o linear. Em todas as bases de dados o valor para o parâmetro de penalização foi  $C = 1.0$ .

#### 4.1.3 Configuração do Naive Bayes

Estudou-se como o tipo de distribuição assumida por cada atributo influenciou o desempenho no método *Naive Bayes*. Os resultados são mostrados na Tabela 4.3 e evidenciam que a melhor escolha para o tipo de distribuição é bastante dependente da base de dados com a qual estamos trabalhando. Mostra-se também o desempenho do modelo induzido no conjunto de validação.

Tabela 4.4: F-measure Média por Modelo

Algoritmo	Sen	J - FT	J - Lic	J - Comp
ANN	0.62	0.76	0.85	0.74
Naive Bayes	0.28	0.56	0.76	0.65
Random Forest	0.70	0.73	0.85	0.76
Regressor Linear	0.75	0.80	0.86	0.77
SVR	0.79	0.76	0.82	0.70
ZeroR	0.61	0.64	0.70	0.60

## 4.2 Desempenho dos Modelos de Aprendizagem de Máquina

Apresenta-se nesta seção os desempenhos dos vários modelos induzidos para, em seguida, discutir os resultados obtidos. Na Tabela 4.4 são mostrados os desempenhos para as bases de dados dos alunos jovens da FT, alunos jovens da licenciatura, alunos jovens da computação e alunos seniores. O melhor desempenho, para cada base de dados, é mostrado em vermelho. O valor de *F-measure* mostrado corresponde a média dos valores de *F-measure* obtidos semestre a semestre.

Por questões de legibilidade, os nomes das bases de dados que aparecem na Tabela 4.4 foram encurtados. Seu significado é apresentado a seguir:

- **Sen:** Base de Dados dos Alunos Seniores.
- **J - FT:** Base de Dados dos Alunos Jovens da FT.
- **J - Lic:** Base de Dados dos Alunos Jovens da Licenciatura.
- **J - Comp:** Base de Dados dos Alunos Jovens da Computação.

Os resultados mostram que, de forma geral, os algoritmos de aprendizagem de máquina conseguem ter um resultado melhor que o ZeroR (à exceção do *Naive Bayes*). O melhor desempenho dos modelos em relação ao ZeroR era esperado, já que o ZeroR é bastante simples, costuma ser usado como *baseline* e não utiliza a informação de entrada que dispõe. Atribui-se o mau desempenho do *Naive Bayes* ao fato de os atributos passados não poderem ser considerados condicionalmente independentes, premissa admitida para o uso do algoritmo.

Por fim, deve-se ressaltar o bom desempenho obtido pelo regressor linear. Tal técnica obteve o melhor desempenho em três das quatro bases de dados (perdendo apenas para a SVR na base de dados dos alunos seniores) e obteve um valor de *F-measure* médio em

torno de 0.795. Isso está de acordo com a teoria de aprendizagem de máquina, que afirma que modelos lineares não são propensos à *overfitting* e que são boas alternativas iniciais em geral [10].

# Capítulo 5

## Conclusão e Trabalhos Futuros

Neste capítulo, apresenta-se a conclusão do trabalho e apontam-se possíveis continuações da pesquisa.

### 5.1 Conclusão

Considerando a idade do aluno, os dados foram divididos em alunos jovens (ingressaram com até 30 anos) e alunos seniores (ingressaram com mais de 30 anos), resultando nas bases: alunos seniores, alunos jovens da FT, alunos jovens da licenciatura, alunos jovens da computação. As diferentes distribuições dos atributos nessas bases de dados foi estudada, incluindo a quantidade de alunos que forma, evade e migra.

Atributos considerados são: sexo, idade, cotista, curso, forma de ingresso, taxa de aprovação, taxa de trancamento, índice de rendimento acadêmico (IRA), taxa de melhora acadêmica, créditos integralizados, taxa de aprovação em disciplinas difíceis, estar em condição e posição no ranque. Também foram induzidos modelos com cauda que consideravam como atributo a estimativa do modelo no semestre anterior. Não houve diferença estatística significativa entre os modelos com e sem cauda.

Os melhores resultados foram obtidos com regressão linear multivariada: alunos jovens da FT (*F-measure* de 0,8), alunos jovens da licenciatura (0,86), alunos jovens da computação (0,77) e alunos seniores (0,75). Nesse último caso, a SVR obteve *F-measure* de 0,79.

O algoritmo de aprendizagem de máquina regressor linear obteve um bom desempenho na tarefa de prever quais alunos seriam capazes de se graduar, quais evadiriam e quais migrariam. Esse resultado aponta a viabilidade do uso de aprendizagem de máquina para análise preditiva de alunos de graduação em risco de evasão na UnB.

Todos os dados utilizados nessa pesquisa estão disponíveis para o SIGRA. Assim sendo, a metodologia usada nessa pesquisa pode ser aplicada para a indução de modelos predito-

res para outros cursos de graduação da UnB. A metodologia poderia também ser replicada para outras IES, desde que essas tenham um conceito análogo ao do IRA.

## 5.2 Trabalhos Futuros

A continuação natural desse trabalho é a implementação de ações na UnB em relação à evasão com base no indicado pelo sistema previsor. Tais ações devem então ser avaliadas, levando em conta questões como redução obtida no índice de evasão e aceitação por parte dos membros da universidade.

Um outro caminho possível de se seguir é testar o desempenho de tal sistema em outros cursos da UnB. Outro caminho que poderia ser seguido é a melhora do sistema previsor. Para isso, uma opção é a inclusão de novos atributos, como por exemplo a quantidade de disciplinas obrigatórias restantes ou a quantidade de créditos do curso do aluno. Outra opção seria a avaliação do sistema previsor por outras métricas que não a *F-measure*, de modo a verificar se o bom desempenho é consistente. Por fim, poderia-se estudar o intervalo de confiança da medida *F-measure* através da utilização da distribuição  $\beta$ .

# Referências

- [1] Raphael Hoed e Marcelo Ladeira: *Cursos superiores que requerem maior capacidade de abstração algorítmica e conhecimento matemático apresentam maiores taxas de evasão? Um estudo dos cursos de Computação no Brasil*. Relatório Técnico. 1, 4, 14, 15
- [2] Raphael Hoed e Marcelo Ladeira: *Sobrevivência dos alunos de cursos superiores de computação: um estudo de caso na Universidade de Brasília*. Relatório Técnico. 1, 14, 15
- [3] [http://www.correiobraziliense.com.br/app/noticia/cidades/2015/10/10/interna\\_cidadesdf,501999/evasoes-na-universidade-de-brasilia-causam-prejuizo-de-r-95-mi.shtml](http://www.correiobraziliense.com.br/app/noticia/cidades/2015/10/10/interna_cidadesdf,501999/evasoes-na-universidade-de-brasilia-causam-prejuizo-de-r-95-mi.shtml). Acessado em 7 de Julho de 2017. 1
- [4] Latiesa, M: *Estudio longitudinal de una cohorte de alumnos de la universidad autónoma de madrid. análisis de la deserción universitaria*. M. LATIESA RODRÍGUEZ (comp.), Demanda de educación superior y rendimiento académico en la Universidad, CIDE, Consejo de Universidades, Madrid, página 437, 1986. 1
- [5] Evasão, Comissão Especial de Estudos de: *Diplomação, retenção e evasão nos cursos de graduação em instituições de ensino superior públicas*, 1996. 4
- [6] <http://www.unb2.unb.br/administracao/decanatos/deg/downloads/index/guiacalouro.pdf>. Acessado em 7 de Julho de 2017. 5, 6, 14
- [7] Chapman, Pete, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer e Rudiger Wirth: *Crisp-dm 1.0 step-by-step data mining guide*, 2000. 6, 7, 8
- [8] Kelleher, John D, Brian Mac Namee e Aoife D’Arcy: *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT Press, 2015. 7, 9, 12
- [9] Noether, Gottfried E: *Why kendall tau*. Teaching Statistics, 3(2):41–43, 1981. 7
- [10] Abu-Mostafa, Yaser S, Malik Magdon-Ismail e Hsuan Tien Lin: *Learning from data*, volume 4. AMLBook New York, NY, USA:, 2012. 8, 10, 11, 48
- [11] Mitchell, Tom M: *Machine learning*. 1997. Burr Ridge, IL: McGraw Hill, 45(37):870–877, 1997. 9

- [12] Ho, Tin Kam: *Random decision forests*. Em *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, páginas 278–282. IEEE, 1995. 9
- [13] Silver, David, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot *et al.*: *Mastering the game of go with deep neural networks and tree search*. *Nature*, 529(7587):484–489, 2016. 11
- [14] Silva, Hadautho Roberto Barros da e Paulo Jorge Leitão Adeodato: *A data mining approach for preventing undergraduate students retention*. Em *Neural Networks (IJCNN), The 2012 International Joint Conference on*, páginas 1–8. IEEE, 2012. 14
- [15] Kinnunen, Päivi e Lauri Malmi: *Why students drop out cs1 course?* Em *Proceedings of the second international workshop on Computing education research*, páginas 97–108. ACM, 2006. 14, 15
- [16] Yu, Chong Ho, Samuel DiGangi, Angel Jannasch-Pennell e Charles Kaprolet: *A data mining approach for identifying predictors of student retention from sophomore to junior year*. *Journal of Data Science*, 8(2):307–325, 2010. 14, 15
- [17] Xie, Yaya, Xiu Li, EWT Ngai e Weiyun Ying: *Customer churn prediction using improved balanced random forests*. *Expert Systems with Applications*, 36(3):5445–5449, 2009. 15
- [18] Domingos, Pedro: *A few useful things to know about machine learning*. *Communications of the ACM*, 55(10):78–87, 2012. 41
- [19] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot e E. Duchesnay: *Scikit-learn: Machine learning in Python*. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 42